NASA Technical Paper 2231

October 1983

F100 Multivariable Control Synthesis Program

Computer Implementation of the F100 Multivariable Control Algorithm

James F. Soeder

NASA

25th Anniversary
1958-1983

NASA
Technical
Paper
2231

1983

F100 Multivariable Control Synthesis Program

Computer Implementation of the F100 Multivariable Control Algorithm

James F. Soeder

Lewis Research Center
Cleveland, Ohio

NASA
National Aeronautics
and Space Administration

**Scientific and Technical
Information Branch**

Summary

Recent aircraft operational requirements have dictated the development of today's sophisticated turbofan engines. Classical control synthesis techniques have worked for older, simpler engines. However, a need exists for a control synthesis procedure that can account for multiple loop interactions and use them to optimize engine performance. One approach to solving the multivariable control problem is to apply optimal control theory. The linear quadratic regulator is one specific area of optimal control theory that has been applied to the turbine engine control problem. The F100 Multivariable Control Synthesis Program is one of these applications. This program is a contracted effort sponsored jointly by the Air Force Aeropropulsion Laboratory and the NASA Lewis Research Center. It is aimed at extending earlier linear quadratic regulator work in order to accomplish the design, evaluation, and testing of a practical multivariable control for the F100 turbofan engine.

This report describes in detail the multivariable control algorithm and its control software implementation for evaluation with a simulation of the engine. In addition, the sensor and actuator failure logic, along with other modifications of the control software to allow it to run with a real engine, is discussed. The results indicate that a modern multivariable control can be programmed on a modest commercial minicomputer and meet update time (sampling interval) and memory requirements. In addition, this software, with minor modifications, can be successfully run with the engine in a research test cell. Recommendations concerning future engine control software development are presented.

Introduction

Over the past several years, aircraft operational requirements have dictated the development of gas turbine engines having increased performance capabilities over a wider operating envelope. These development efforts have resulted in today's sophisticated turbofan engines and will undoubtedly lead to increasingly complex engines. For example, future variable-cycle engines may incorporate variable fan, compressor, turbine, and exhaust nozzle geometry to optimize overall engine performance (ref. 1). The trend toward more complex engines has resulted in additional requirements for the control system. Controls for future engines will have to measure more engine variables (perhaps 10 to 20) to control both the engine fuel flows and the variable geometry. For greater control accuracy and response the closed-loop control is replacing the scheduling (open loop) control used on older engines.

Classical control design (synthesis) techniques, which involve the design and evaluation of single-input, single-output control loops, have worked for the older, simpler engines. However, such techniques are cumbersome and time consuming when applied to the multivariable control problem because the inherent loop interactions are not easily considered in the classical design process. A need exists for a more suitable control synthesis procedure—that is, one that can account for these loop interactions and possibly make use of them to optimize engine performance.

One approach to solving the multivariable control problem is to apply modern (optimal) control theory. This approach appears to be suited to the engine control problem because of the emphasis on maintaining optimum engine performance in the presence of a wide variety of external disturbances (i.e., aircraft maneuvers, horsepower extraction, etc.). The linear quadratic regulator (LQR) is one specific area of modern control theory that has been successfully developed and applied to a wide variety of linear, multivariable control problems (ref. 2). There have also been some initial research and development efforts aimed at applying LQR theory to the design of controls for a nonlinear engine process (ref. 3 to 7). These efforts, however, have been limited to engine control over a narrow operating range (usually sea-level, static, standard-day conditions).

The F100 Multivariable Control Synthesis (MVCS) Program, a contracted effort sponsored jointly by the Air Force Aeropropulsion Laboratory (AFAPL) and the NASA Lewis Research Center, was aimed at extending the earlier LQR work to the design and testing of a "practical" multivariable control for a state-of-the-art turbofan engine—that is, a control capable of operating an engine over its entire operating envelope. The engine selected for the MVCS program was the Pratt & Whitney F100-PW-100(3) afterburning turbofan. In addition to the design of a control for the F100 engine, the MVCS program goals also include the identification of advantages and disadvantages of the LQR method for designing engine controls; the evaluation of the control design by using a real-time, hybrid computer simulation of the engine; and finally, the demonstration of a multivariable control of the F100 engine in a NASA altitude test facility.

To achieve these objectives, a program was formulated with the following division of responsibilities: The Air Force provided the contract vehicle for the engine manufacturer and for the controls research organization and was therefore responsible for monitoring the activities of these two contractors. Pratt & Whitney as the engine manufacturer had the prime responsibility for defining the F100 engine performance, both steady state and transient, by means of a nonlinear, digital computer simulation (CCD1103-1.0) of the engine. This digital engine simulation also formed the basis for a set of linear engine models that Pratt & Whitney supplied to Systems Control, Inc., the controls research organization, for use in the LQR design procedure (ref. 8). Systems Control, Inc., had the prime responsibility for designing the

multivariable control logic and demonstrating that this logic can successfully control the Pratt & Whitney simulation of the engine (ref. 9).

The Lewis Research Center was responsible for developing a real-time hybrid computer simulation of the engine (ref. 10), for providing the real-time software implementation of the control algorithms, for evaluating the control by using hybrid real-time engine simulation, and finally for planning and conducting the engine tests. A hybrid simulation evaluation has three distinct advantages. First, the control software that will be used in the actual engine test can be verified. Second, once the simulation and control software has been debugged, steady-state and transient data can be generated quickly and easily. This makes it possible to evaluate more data points and to uncover more problems than would be economically feasible on an all-digital, non-real-time simulation of the engine and the control. Third, the control can be run in the same way as it would be when used with the real engine in the test cell, and therefore more hands-on experience can be gained with the control than is possible with the digital simulation deck.

This report describes (1) the F100 turbofan engine, (2) the facilities and equipment used for the control evaluation, (3) the multivariable control in detail, (4) the real-time software implementation of the multivariable control used in the hybrid simulation evaluation, and (5) logic additions and modifications incorporated for the full-scale engine tests. Finally, conclusions and recommendations with regard to future control development are presented. The results of the control evaluation with the hybrid simulation are documented in a NASA report (ref. 11), and the results of the full-scale engine tests are documented in reference 12.

Description of Controller Evaluation Apparatus

F100 Turbofan Engine

The engine selected for the MVCS program was the Pratt & Whitney F100-PW-100(3) afterburning turbofan, which is representative of current high-techology engines. A schematic drawing of this engine is shown in figure 1; and a cutaway drawing, in figure 2. The F100 is a twin-spool, axial-flow turbofan with a bypass ratio of 0.7 at sea-level, static, standard-day conditions. A single inlet is used for fan airflow and engine core airflow. Airflow leaving the fan is separated into two flow streams: one stream passing through the core, and the other stream passing through the annular fan duct. A three-stage fan is driven by a two-stage, low-pressure turbine. A two-stage, high-pressure, air-cooled turbine drives the 10-stage compressor. The fan has variable-trailing-edge compressor inlet guide vanes (CIVV's). These inlet guide vanes are positioned to improve inlet distortion tolerance and fan efficiency. The compressor has rear rows of variable stator vanes (RCVV's). These vanes are positioned to improve starting and to provide good compressor operating characteristics. Airflow bleed is extracted at the compressor exit and is discharged through the fan duct for starting. Bleed is also extracted to satisfy installation requirements and to provide turbine cooling. The main combustor consists of an annular diffuser and a chamber with 16 fuel nozzles. The engine core and fan duct streams combine in an afterburner that consists of a diffuser and five concentric fuel manifolds. The afterburner discharges through a variable convergent-divergent nozzle. The exhaust nozzle is a balanced-beam design with an activated divergent flap. The variable nozzle geometry provides nearly optimum nozzle area, expansion ratio, and boattail drag through the operating range.

The current bill-of-material (BOM) control is hydromechanical with an engine-mounted digital electronic supervisory control. The hydromechanical fuel control system (1) meters fuel to the main combustor, (2) positions the compressor vanes to improve starting and high-Mach-number characteristics, (3) meters fuel to the five augmentor zones, and (4) controls the nozzle area so as to maintain the desired engine airflow during augmented operation. The electronic supervisory control (1) positions the inlet guides vanes for best fan efficiency, (2) trims the main-combustor fuel flow to satisfy engine temperature and pressure limits, (3) trims the nozzle area to satisfy engine airflow requirements, and (4) limits minimum power lever angle as a function of Mach number and combustor pressure.

Hybrid Computer Facility

Figure 3 shows the interconnection of the computers in the NASA hybrid simulation facility used in evaluating the multivariable control. An Electronic Associates model 690 hybrid computer was used to implement the equations describing the F100-PW-100(3) engine. The sensed variables from the simulation were fed into an Electronic Associates model TR-48 analog computer, which contained the simulations of the engine sensors. The sensor outputs were fed into an SEL 810B minicomputer, which contained the real-time software implementation of the control algorithm along with the engine actuator simulations. The outputs (or actuator commands) of the SEL 810B digital computer were then fed back into the hybrid-computer engine simulation to close the control loop.

In addition to doing the control calculation and actuator simulations, the SEL 810B also acted as a data collector for the simulation. This was done by storing the time histories of variables of interest in the computer's spare core memory. The data were then dumped from the computer memory through a Tektronix 4010 terminal and onto a Tektronix 4922 floppy disk. The data from the floppy disk were then transmitted via telephone line

to the Lewis IBM 360 computer for remote processing and plotting. The various computers and peripherals are shown in figures 4 to 7.

Real-Time Engine Simulation

A real-time, hybrid-computer simulation of the F100-PW-100(3) turbofan engine was developed by Lewis to support controls research programs involving that engine (ref. 10). The simulation has both wide-range, steady-state and transient computing capabilities. The mathematical model describing the performance of the F100 engine was patterned after the CCD1103-1.0 digital simulation (ref. 8). Wide-range, overall performance maps for the engine's rotating components (i.e, fan, compressor, and turbines) were used to provide wide-range, steady-state accuracy. Factors such as fluid momentum, mass and energy storage, and rotor inertias were included to provide transient capability.

Figure 8 is a computational flow diagram of the real-time simulation. It can be seen that the mathematical model consists of a number of individual elements and their related volumes, each of which requires a number of input variables and generates one or more output variables.

The equations describing the F100 mathematical model were implemented on the Lewis hybrid computing system. The Electronic Associates model 690 hybrid computer consists of a model 640 digital computer, a model 680 analog computer, a model 681 analog computer, and an interface unit that allows communication between the individual computers. The digital portion of the hybrid computer was used primarily to perform the bivariate function generation associated with modeling the performance of engine components. In addition to function generation, the digital computer was used for computing the fan and compressor surge margins and the engine net thrust. The digital portion of the simulation was structured with an update time that provided essentially accurate, real-time dynamics. The remaining calculations were performed on the analog computers. The analog computers provided continuous integration with respect to time, multiplication, division, and univariate function generation. The two analog computers were fully utilized. For example, the full complement of 54 multipliers and a total of 189 potentiometers were required. The use of peripheral equipment such as X-Y plotters and strip-chart recorders allowed continuous monitoring of computed variables. The analog portion of the simulation was set up and checked out by using the EAI 640 digital computer and some general-purpose software. This system is discussed in more detail in reference 10.

SEL 810B Digital Computer

An SEL 810B digital computer was used to implement the multivariable control. The computer is a two-accumulator, 16-bit machine with an index register. It features a 24-kiloword magnetic core memory with a 750-nsec memory cycle time and a two's-complement, fixed-point hardware multiplication and division circuit. It also has analog-to-digital and digital-to-analog converters for input and output of the control signals. The computer is shown in figure 7. Table I summarizes the specifications of the computer and its peripherals. Although this is a fairly modest computer compared to what is commercially available today, it is believed to be similar in word size, instruction repertoire, and other capabilities to one that will be engine mountable in 1 to 2 years. Therefore it was felt that the SEL 810B would make an excellent vehicle to implement and demonstrate the multivariable control. Further information on the SEL 810B is given in reference 13.

Full-Scale Altitude Test Facility

Figure 9 is a block diagram of the test setup used in the full-scale altitude tests of the multivariable control. These

TABLE I. - MULTIVARIABLE CONTROL COMPUTER SPECIFICATIONS

SEL 810B digital computer
Two 16-bit accumulators Memory specifications: 24-kiloword magnetic core Cycle time, 0.75 μsec Expandable to 32 K Two's complement, fixed-point multiplication and division: Addition time, 1.5 μsec Multiplication time, 4.5 μsec Division time, 8.25 μsec Double-precision arithmetic Infinite indirecting Infinite indexing Direct memory access 28 Levels of vectored priority interrupt 66 Total instructions
Analog acquisition unit[a]
Two multiplexors, sample-and-hold circuit, and analog-to-digital converters 64 Input channels for each multiplexor Input voltage range, ±10 V 12-Bit (plus sign) data resolution (two's complement) Digitizing rate, 50 μsec/sample Percent error with calibration, 0.073
Analog output unit[a]
26 Digital-to-analog conversion channels: Ten 12 bit (plus sign) Sixteen 11 bit (plus sign) Output voltage range, ±10 V Slew rate, 1 V/sec
Tektronix 4010A peripheral system[a]
Tektronix 4010 scope terminal: 1600-Baud capacity RS232 and teletype interface Paper tape reader punch Floppy disk: Store 262 144 bytes/disk (660 m of paper tape) Write speed, 400 bytes/sec

[a]SEL 810B peripherals.

tests were carried out in two laboratories. One was the Propulsion Systems Laboratory, or PSL, which contained the engine, the altitude cell, the research actuators, and the backup control system. The other was the Hybrid Computer Laboratory, which contained the SEL 810B computer, associated operator interface peripherals, and the A/D and D/A converters. The sensed variables and actuator commands that were necessary to control the engine were exchanged between the two facilities via 305-m-long (1000-ft-long) underground cables. In PSL all of the sensed signals were passed through 10-Hz filters to the SEL 810B remote unit, which is shown in figure 10. The remote unit provided signal buffering, patchboard capabilities for ease of signal troubleshooting, and a set of status lights to provide the engine operators with information on the multivariable control's operating status.

Figure 11 shows the F100 engine in the altitude chamber. The altitude chamber is capable of running the F100 engine at flight points throughout its operating envelope. The test tank included a forward bulkhead separating the inlet plenum from the altitude chamber. Conditioned air was supplied to the plenum at the pressure and temperature corresponding to the chosen flight condition. The chamber aft of the bulkhead was evacuated to the desired altitude pressure. The conditioned air flowed from the plenum through a bellmouth inlet (to supply undistorted flow) and finally to the engine compressor face. A valve in the bulkhead allowed some of the air to bypass the engine. This valve was automatically controlled to maintain a constant inlet pressure and a constant ram pressure ratio across the engine during steady-state as well as transient operation. The exhaust from the engine was captured by a collector, extending through the rear bulkhead, to minimize the recirculation of exhaust gases into the PSL altitude chamber. The exhaust, and thus altitude pressure, was controlled by an automatic valve.

Engine Actuators

For the engine to function under the control of a digital computer (SEL 810B), electrohydraulic servosystems had to be added to actuate fuel flow, jet nozzle area, rear compressor variable vanes, compressor-inlet variable vanes, and bleed. The research actuators were all mounted on or near the engine in the test cell. Five electronic panels (fig. 12) located in the PSL control room made it possible to individually switch from the backup system to the research actuators.

Table II is a summary of the type of research actuators used for the control tests and the backup systems used in the event of a problem with the multivariable control. Although each of the actuators was engaged individually, they were wired in such a way that in the event of a research control abort all of the research actuators would revert to their backup modes simultaneously. Individual actuator operation is discussed more completely in reference 12.

Multivariable Control Description

Overview

Figure 13 is an overall block diagram of the multivariable control, showing the overall signal flow and emphasizing the modular nature of the control. Seven modules are included: reference-point schedules, transition control, integral control, linear quadratic regulator, gain control, engine protection logic, and fan turbine-inlet temperature (FTIT) estimator. The overall control mode provided by this configuration is basically proportional plus integral with a feedforward path for fast response. The proportional action is provided by the LQR, which has regulator gains that can affect changes in all of the available control variables. This reduces

TABLE II. - RESEARCH ACTUATORS

Actuator name	Research actuators	Backup system
Fuel flow	Electronic servo in series with unified control	Hydraulically switches to unified control
Rear compressor variable vane (RCVV)	Electronic servo in parallel with unified control	Hydraulically switches to unified control
Compressor-inlet variable vane (CIVV)	Electronic servo modulating standard CIVV actuator piston	Electrically switches servo input to analog computer
Exhaust nozzle	Electronic servo modulating standard air motor actuator	Electrically switches servo input to a fixed value
Bleed flow	Electronic servo modulating a special bleed valve	Electrically switches servo input to zero

deviations in all of the state variables relative to specified reference values. The state, control, and output variables used in the multivariable control are given in table III.

The LQR algorithm used in this study was synthesized by using continuous system design techniques. Since the update time for the total control was less than 10 msec, it was assumed that discrete LQR synthesis techniques would not be necessary.

The integral control provides steady-state trimming of the engine operating point to satisfy performance requirements and engine limits. The integrators are controlled by limit flags generated by either the engine protection logic or the transition control. The engine protection logic limits fuel flow, nozzle area, and geometry excursions to provide safe, stable engine operation.

The steady-state reference values of the state and control variables are scheduled as functions of the pilot-commanded power lever angle (PLA) and the flight condition. The outputs of the reference-point schedules are rate limited in the transition control to prevent excessive deviations, which could saturate the LQR. The transition control provides a transient model for the system to follow.

Because of the nonlinear nature of the engine process, a single set of LQR and integrator gains will not result in satisfactory closed-loop responses at all operating conditions. To provide wide-range operation, the gains are scheduled as functions of the fan-inlet conditions and the transition value of compressor speed. Note that the reference-point and gain-scheduling problems are not addressed in references 3 to 7.

The current F100 control system uses lead compensation to correct for a slow fan turbine-inlet temperature sensor. The F100 MVCS design uses an FTIT estimator during transient operation to predict whether an overtemperature will occur. The estimated value of FTIT is compared with the limiting value and, if required, the integral control downtrims the engine fuel flow before an overtemperature can occur.

The following sections provide a more detailed description of the various elements that make up the multivariable control. Two items should be remembered during the following discussion. First, the control, although modeled in analog fashion, is computed sequentially in the following order: FTIT estimator, reference-point schedules, gain control, transition control, integral control, LQR, and engine protection logic. Second, figure 13 should be referenced frequently to keep in mind the overall signal flow between the modules.

Reference-Point Schedules

The reference-point schedules use TT2MES, PLAMV, SMNMES, and PT2MES to produce steady-state values of the engine states and control outputs throughout the engine operating envelope. (All symbols are defined in the appendix.) Figure 14 is a detailed block diagram of the computation. The reference-point schedules can best be understood by dividing them into three sections: the core states, the fan states, and the fuel flow and limiting logic.

The core-state calculation starts by determining a value for SNCSCI by using TT2MES to generate the speed breakpoint values SN2INT, SN2HI, SN2MID, and SN2LOW. These speed values correspond to PLAMV of 83°, 78°, 36°, and 20°, respectively. The current PLAMV is interpolated between the two appropriate boundary values to get SNCSCI. This value is corrected by using THETA and input to the PBCRV to generate PT4SCI ratioed to PT2MES (P4OPT2). A constant, PBBIAS, is added to allow for differences between the simulation and the real engine. Multiplying PT2MES by P4OPT2 gives an unlimited value for the transition combustor pressure, PT4SCI. If PT4SCI is greater than the normal case, PT4SCI and SNCSCI are passed to the limiting logic. However, if PT4SCI is less than the normal case, S1, S2, and S3 are switched to allow for a back calculation of SNCCR2, SNCSCI, and PLAMV based on limiting PT4SCI. This is done by computing a PT4SCI/PT2MES ratio with PT4SCI equal to the normal case. The new ratio, PBOPT2, is input to the PBICRV (inverse of the PBCRV) to generate a new SNCCR2. The SNCCR2 is uncorrected to give an SNCSCI that is limited to a maximum value of SN2INT in order to generate the new SNCSCI. The new SNCSCI is also input to inverse interpolation logic with SN2INT, SN2HI, SN2MID, and SN2LOW acting as x breakpoints and their respective PLAMV's from the previous interpolation acting as y breakpoints. A new uptrimmed PLAMV is interpolated and used throughout the rest of the calculations to match the other state output values to the minimum combustor pressure. Whichever SNCCR2

TABLE III. - VARIABLES USED IN MULTIVARIABLE CONTROL

State variables	Control variables	Output variables
Fan speed	Fuel flow	Thrust
Compressor speed	Nozzle area	Fan airflow
Afterburner pressure	CIVV	Combustor temperature
Combustor pressure	RCVV	Fan surge margin
Fuel flow	Bleed flow	Compressor surge margin

value is chosen, normal or uptrimmed, is then used by the TITCRV to compute a value of TITOT2. A bias that is a function of the density parameter DEN is added to TITOT2 to take into account Reynolds number effects in the upper left corner of the operating envelope. A multiplication by TT2MES results in a value of FTITSH. For the actual engine tests it is necessary to bias the schedule below a PLAMV of 78° and gradually to reduce the bias to one-half its value at a PLAMV of 83°. This results in the final value of FTITSH.

In the fan-state calculation the unlimited scheduled airflow, WDOTS, is calculated as a function of PLAMV and TT2MES. WDOTS is modified at low PLAMV and low SMNMES by WAMNCV and associated PLAMV interpolation logic. This is done to provide a better match to the BOM control at idle, sea-level, static (SLS) conditions. The resulting WDOTSI is minimum and maximum limited by the WACHI and WACLO curves, respectively. These curves enforce the F-15 inlet airflow limit corridor and this gives the final scheduled airflow, WDOTSH. DP25SI is now generated by using WDOTSH and DP25CRV. DP25SI is biased at high PLAMV and at high TT2MES by D25DCRV and this gives the final value for DP25SH. WDOTSH is input to SNFCRV, which results in SNFSCR. This value is biased to make allowance for simulation engine mismatch and then uncorrected, giving an unlimited scheduled value for fan speed. Finally the scheduled value for afterburner manifold pressure, PT6SCH, is computed by inputting WDOTSH and TT2MES to PT6CRV. This value can be biased for engine and simulation variations. In addition, at high inlet temperatures the PT6SCH value is multiplied (rotated) by a factor proportional to TT2MES.

The limiting logic starts by imposing limits on previously computed but unlimited state values. PT4SCI is minimum and maximum limited and thus becomes PT4SCH. SNFSCI and SNCSCI are both limited as a function of TT2MES by the FANMAX and SMNCRV curves, respectively, and thus become SNFSCH and SNCSCH. FTITMX is generated as a function of TT2MES by the FTICRV curve. Note that the output of the FTICRV curve can be biased by FTMXB to allow the engine to run to any desired fan turbine inlet temperature. The scheduled fuel flow calculation starts by computing corrected fuel flow CWF as a function of WDOTSH and SMNMES. The output of the CWFCRV curve is then rotated by WFROT. This allows easy compensation for differences between the simulation and the engine. The result, CWFMB, is uncorrected by using DELTA and THETA and then compared with minimum fuel flow, which is a function of PT4EST. The result of this comparison, WFSCI, is compared with an absolute maximum, WFMAX, and then with the minimum, WFMIN; WFSCI finally results in scheduled fuel flow WFMBSH. WFMIN is computed as a function of TT2MES and SMNMES by using the WACMIN and CWFCRV curves. The scheduled exhaust nozzle position AJACH is set to a nominal value. However, as TT2MES, and hence airflow, increases, the nozzle is scheduled further open. Finally, three limit flags are set on the basis of the sensed values FTITEST and PT4EST. First, MTFFLG is set to 1 if FTITEST is above FTITMX minus 4. In addition, MPBHI or MPBLO is set to 1 if PT4EST is above PBMAX minus 4.5 or below PBMIN minus 2.0, respectively.

Gain Control

Figure 15 is a detailed diagram of the gain control. The gain control schedules the gains for the LQR and integral control modules of the multivariable control. This is necessary to provide satisfactory response throughout the flight envelope. Figure 16 shows the final total gain matrix, with appropriate zeroed elements, that is output by the gain control. The first five columns of the matrix correspond to the LQR gain matrix, and the last eight columns represent the integral control gain matrix. The gain control generates this final matrix by interpolating the nonzero elements of six like matrices: four describing high-power operation, and two describing low-power operation.

This interpolation is done in the following manner: the value of DEN is examined to determine which two high-gain numbers and hence which two of the four high-power gain matrices are operative at the present flight condition. Once determined, the value of HMUL is computed by interpolating DEN between the two high-gain values. Similarly, for the low-power gain matrix, DEN is limited between LG1 and LG2, resulting in GPARM. GPARM is then interpolated between LG1 and LG2 limits, resulting in LMUL. HMUL and LMUL are then used to interpolate each nonzero element of the high- and low-power matrices, resulting in the CHXX and CLXX values, respectively. To compute the final matrix gains, TT2MES is input into the XN2HCV and XN2LCV curves, resulting in upper and lower speed breakpoints. The transition value of compressor speed, SNCTR, is interpolated between these two points, resulting in FMUL. FMUL is then used to generate the final matrix values, CX, by using the CLXX's and CHXX's. FMUL is also used to generate AJPARM, the use of which is explained later in the section Engine Protection Logic.

Transition Control

Figure 17 is a detailed block diagram of the transition control. The transition control provides a rate-limited trajectory for the state and control outputs of the reference-point schedules. This control is necessary to prevent large regulator state deviations and hence

saturated control outputs during a large excursion from one operating point to another.

The calculation starts by determining rates of movement for each of the scheduled states and controls. The four speed breakpoints—SN2INT, SN2HI, SN2MID, and SN2LOW—are used. These are the same values that were calculated in the reference-point schedules to compute the SNCSCI parameter. These four speeds define four x breakpoints in the DELTRN schedule. Associated with each x breakpoint are nine y values, one for each of the states and controls that is to be transitioned. Therefore DELTRN represents a set of nine curves each with the same four x breakpoints and y values corresponding to the transition rate of the respective state or control. The transition value of speed SNCTR is then input to the DELTRN schedule nine times to determine the raw values of the five state rates: PT6RAT, PT4RAT, FTIRAT, FANRAT, and COMRAT; two control rates: WFRATE and AJRATE; and two control steps: AJDEL and WFDEL. These raw rates are modified as a function of DEN by either PARM1 or PARM2. These parameters allow modification of the rates as a function of flight condition. Next, PLAMV is limited by the PLALMT schedule, resulting in PLAMP. The value of PLAMP from the previous iteration is subtracted from PLAMP to get PLAD. If the absolute value of PLAD is greater than the 20-deg/sec deadband in the PLARAT schedule, a nonzero value for F is determined by the PDIFF schedule. The maximum and minimum values of PDIFF are determined as a function of PLAD. The scheduled and transition values of high rotor speed, SNCSCH and SNCTR, respectively, are subtracted to generate DIFF. The factor G is then generated by dividing DIFF by 1000 rpm, subtracting it from 1, and taking the absolute value of the result. The variables F and G are multiplied to yield FG. FG then multiplies the AJDEL and WFDEL factors. Close examination of the FG generating logic shows that the only time FG has a nonzero value is when the PLAMV is chopped from above 67°. This is done because lower rates are chosen at high-power design points than at midpower design points to implicitly limit temperature overshots. However, during deceleration from midpower points, the respective rate trajectories will follow the acceleration path in reverse. By adding the FG parameter and hence the step in the upper portion of the power range, fast decelerations and temperature-limited accelerations are produced.

If the absolute value of DIFF is bigger than 50 rpm, the value of MTRAN is set to 1. MTRAN = 1 indicates that the scheduled engine reference point has moved and that the transition control is active and moving the reference point. If the transition control is active, MAJFLG, MRCFLG, MCVFLG, and MBLFLG are all set to 1 to freeze the integral control. This is done to prevent too much erroneous value from building up on the integrator outputs during a large transition and thus resulting in overall underdamped or unstable behavior of the control. The fuel flow limit flag MWFFLG is set to 1 (freezing the fuel flow integrators) only if the MTRAN flag is on and the system is not on a combustor pressure or fan turbine-inlet temperature limit.

The transition values (SNFTR, SNCTR, etc.) are generated from the scheduled values (SNFSCH, SNCSCH, etc.) by putting each of the scheduled variables through a rate-variable, first-order lag. Each scheduled variable has its respective transition value subtracted from it. This error is input into RATCV to determine what fraction of its respective rate from the curve interpolation will be used in the next integration time step. The integration is performed by adding the necessary DELTA. Repeating this process for each of the scheduled variables generates their respective transition values. Note that the transition rates are frozen by switch S4 if the following condition exists: the difference between the limited and unlimited fuel flows multiplied by the rate being considered is greater than zero and the fuel flow limit flag MWFFLG is 1. If this condition exists, all of the transition values (SNFTR,SNCTR, etc.) are frozen at their last value. This allows the control to run when it is against the limits imposed by the engine protection logic.

Integral Control

The purpose of the integral control is to eliminate the steady-state error or "hang off" that would be generated from the use of only proportional action as provided by the LQR. This error results because of approximations in the scheduling algorithm, engine-to-engine variation, and aging effects, which cause the reference-point schedules to not always specify an exact engine equilibrium.

The integral control calculation begins by generating a scheduled value for RCVV as a function of SN2EST and T25MES by using the RCVCRV curve and limiting the output to between +6° and −40° (fig. 18(a)). The scheduled value of CIVV is generated as a function of FANCOR and limited to between 0° and −25°. Next the integrator errors (EDP25, ECIVV, ERCVV,EBLC, ESN1,EFTIT, EPBMX, and EPBMN) are computed and appropriate deadbands are added. A ±500-rpm limit is imposed on the low rotor speed error ESN1. This prevents large scheduling errors during transients from causing the fuel flow integrator to wind up and result in underdamped behavior. The integrator input errors are trimmed in various combinations depending on the state of the engine. As a general rule the first four errors—EDP25, ECIVV, ERCVV, and EBLC—are always trimmed. Of the last four errors—ESN1, EFTIT, EPBMX, and EPBMN—only one is trimmed at a time. The error that is used is dictated by the presence or absence of the effect of combustor pressure on fan turbine-inlet temperature limits. This choice is made by

an algorithm that sets one integrator limit flag (either LMT5, LMT6, LMT7, or LMT8) depending on the state of MPBHI, MPBLO, and MTFFLG. If more than one limit is in force, for example, fan turbine-inlet temperature (MTFFLG) and low combustor pressure (MPBLO), the fan turbine-inlet temperature always takes precedence. Furthermore, if no limits are in force, the low rotor speed error ESN1 is trimmed.

Schematic output diagrams for each integrator are shown in figure 18. The diagrams show the control logic implementation of the matrix structure in figure 19. Inspection of the integrator logic diagram shows that each one is implemented by using simple rectangular integration. The fuel flow integrator output IWFMB is driven by EDP25 if the exhaust nozzle is not saturated (e.g., MAJFLG set) and by either EFTIT, ESN1, EPBMX, or EPBMN depending on which limit flag is set. The integrator output is limited to prevent a large error from building up on the integrator as a result of engine-schedule mismatch. This would cause the control to behave poorly during a transient. The inputs are controlled somewhat differently for the area integrator, IDP25. EDP25 unconditionally drives IDP25. One of the four limit errors drives it only if a fuel flow limit is not in force. The absolute value of IDP25 is limited for the same reason that the IWFMB integrator output is limited. The compressor-inlet variable vane integrator ICIVV can be driven conditionally by three errors: by EDP25 if the exhaust nozzle is not saturated, by ERCVV if the fuel flow is limited and the high rotor speed is currently being trimmed, or by the ECIVV error itself. The CIVV, RCVV, and bleed integrators are trimmed at all times by ECIVV, ERCVV, and EBLC, respectively. These are the errors between scheduled and command values for the three integrators. These integrators have the primary responsibility of making sure that the commanded positions of these actuators are the same as the BOM-scheduled positions in steady state.

The last component of the integral control is the integrator hold logic. Each of the integrator outputs (IWFMB, IDP25, IRCVV, ICIVV, and IBLC) are run through this logic. If the integrator limit flags (MWFFLG, MAJFLG) are off, the integral values (IWFMB, IDP25, etc.) pass through unchanged. However, if the respective limit flag is on, the integrators are only permitted to run in the direction of decreasing absolute value. This prevents the integrators from erroneously winding up when the control runs to a limit and also permits the integrators to unwind from the limits as conditions change.

LQR Control

Figure 20 is a detailed schematic diagram of the linear quadratic regulator portion of the multivariable control. The purpose of the LQR is to provide a set of proportional gains (fig. 21) that act through all of the control variables to minimize the state deviations and thereby provide good control for small perturbations or disturbances. The LQR calculation begins by computing the five state deviations DN1, DN2, DPT6, DWF, and DPB. These are equal to the respective sensed values minus the scheduled values. The nonzero LQR matrix values, computed by the gain control, are then used in a standard matrix multiplication to compute the LQR output contribution to each of the control variables. The LQR output values of the CIVV's and RCVV's are limited to keep the geometry motion within the regions of accurate modeling and to avoid aerodynamic flutter boundaries. The LQR contribution to the exhaust nozzle area output is limited because of surge margin limitations on the fan. Finally the total unlimited control outputs are formed by adding the respective transition control values (or simply the scheduled values in the case of the RCVV's and CIVV's), the integral control values, and the LQR control values for each of the control outputs.

Engine Protection Logic

Figure 22 shows detailed diagrams of the engine protection logic. The purpose of this logic is to limit the control output values to safe operating ranges throughout the entire engine flight envelope. Furthermore, if one of the control outputs is on a limit, the logic sets the respective control saturation flag (MAJFLG, MBLFLG, MCVFLG, MRCFLG, or MWFFLG) to stop that particular control output's trim integrator. The fuel flow limiting logic computation begins by computing WFPBMX from SN2EST by using the WFPBCRV curve. This curve is the WF/PB acceleration limit curve from the BOM control. The multivariable control, like the BOM control, uses this schedule only as a last resort in fuel flow limiting since the transition control defines the transient trajectory. Note that in the BOM control the governor droop slopes are a function of T25MES and power lever angle. This temperature and power bias provides the acceleration limiting; therefore the acceleration schedule is only a worst-case limit. For this reason the WFPBCRV provides only worst-case limiting and is therefore a function of only mechanical high rotor speed. The output of the WFPBCRV curve WFPBMX is added to WOOFBI, which biases the curve either to allow rate-limited accelerations or to eliminate limiting entirely in order to allow for high rate accelerations. The result of this addition is then multiplied by PT4EST and compared with WFMAX in order to generate the fuel flow upper limit. The lower limit is computed by using PT4EST and multiplying it by a standard WF/PB minimum ratio, WFPBMN. This result, WFMNN, is compared with WFMIN, which was generated in the reference-point schedules to give a final value for minimum fuel flow. The maximum and minimum are compared with the LQR

output WFMBCL; and the final fuel flow output WFCOM is computed.

The final RCVV output is computed as a function of COMCOR by using the RCMNCR and RCMXCR curves to determine the minimum and maximum vane positions, respectively. The RCMXCR curve defines the limit boundary for the fourth-stage compressor stall flutter region, and the RCMNCR curve defines the boundary for the sixth-stage compressor choke flutter region. The LQR output RCVVCL is compared with the maximum and minimum to generate the final output RCVVCM.

The final exhaust nozzle area is computed as a function of AJPARM by using the AMNCV and AMXCV curves to determine the minimum and maximum nozzle areas, respectively. These curves lock the nozzle area at idle and set the maximum and minimum at military power. The AMXCV curve provides a maximum nozzle limit to prevent overspeeds, and the AMNCV curve provides a minimum nozzle limit to insure that a minimum surge margin is maintained. The AJPARM parameter is computed in the gain control as a function of SNCTR. This allows the limits on the exhaust nozzle to be enforced gradually during a deceleration and thereby provides another degree of freedom in the control during this time. The LQR output AJCL is compared with these maximums and minimums to generate the final output.

The CIVV limiting logic starts by computing N1P2BI as a function of PT2MES. N1P2BI is added to FANCOR and input into the CVMXCV curve. The output of this curve is the CIVV maximum limit, which defines the edge of the fan flutter boundary. The LQR output CIVVCL is compared with the above maximum and with a fixed minimum to generate the final output.

The LQR output for the bleed BLCCL is limited to a maximum, BLCMX, and a minimum, BLCMN, which are percentages of core airflow. Although a negative bleed is not physically possible, BLCMN is slightly negative. Therefore the bleed trim integrator runs slightly negative. This provides a quiet command to the valve, keeping it closed in steady state. Thus noise is suppressed on the bleed command, which is coupled in through the sensed signals that are fed into the LQR.

FTIT Estimator

Figure 23 is a detailed block diagram of the FTIT estimator. The purposes of the estimator are to filter the FTIT thermocouple signals and to provide phase lead compensation of the slow FTIT signal. The estimator uses the difference between WFCOM and WFMBTR rescaled by the output of the FTWTCV curve to give the lead capability. The result is then added to the outputs of two cross-coupled integrators X1 and X2 to generate Z1. Likewise, FTITMES and FTITSH are subtracted and the result added to cross-coupled integrator outputs X1 and X3, yielding Z2. The outputs Z1 and Z2 are then fed back into the integrator inputs through multiplicative constraints K. The X1, X2, and X3 integrator outputs are the internal states of the estimator. The X1 output is the temperature state that is added to the feedforward terms FDWF and FTITSH to produce the final estimated value of fan turbine-inlet temperature FTITEST. The X2 state is the bias state. The integrator is driven by Z1 multiplied by K21, where K21 can be thought of as the disbelief in the fuel flow estimate of FTIT. Then X2 is fed back to null out the effect of fuel flow in the estimation of FTIT in steady state. The X3 state is the correlated noise state and is driven by Z2 multiplied by K32. The K32 factor can be thought of as the measure of disbelief in the FTIT measurement. Finally the X1 integrator is driven by Z1 and Z2 multiplied by K11 and K12, respectively. The K11 and K12 reflect respective beliefs in the temperature and fuel flow measurements. The feedforward estimator contribution FDWF uses the fuel flow difference FFWD to produce added lead during a transient. Although this lead term produces a small offset in the estimated value, it is in the pessimistic direction and results in negligible loss of performance. In a transient the MTRAN flag sets the bias state X2 equal to zero and the noise state X3 equal to DELTM. The temperature state Z1 is also set to zero and thereby sets the FTITEST value equal to the scheduled value FTITSH. This is done to avoid startup transients and the necessity of reinitializing of the algorithm.

Software for the Multivariable Control

Overview

In programming the multivariable control the stated objectives were to realize a 10-msec update interval and to have the resulting object code fit into the memory of the computer used. Accomplishing these objectives required the use of assembly language programming and scaled fraction arithmetic. The use of assembly language was necessitated because using the SEL 810B's Fortran compiler to process the control algorithm from the digital deck would have resulted in an amount of code that would be too large to execute in 10 msec. The use of scaled-fraction arithmetic was dictated because the SEL 810B has only a fixed-point hardware addition-and-subtraction-circuit and a fixed-point hardware multiplication-and-division circuit. Therefore fixed-point numbers can be manipulated quickly. However, because of their nature, fixed-point numbers are difficult to use. The reason is that they are defined as follows:

$$\frac{\text{Number of engineering units}}{\text{Number of machine units}}$$

Since the number of machine units is fixed (i.e., because the computer word size is fixed) the maximum size of any

particular variable is limited by this value. Therefore one must pick scale factors carefully to insure that they are large enough to accommodate the maximum values that the variable will reach but that they are not so large that the variable scale becomes too coarse.

From the preceding discussion, two problems emerge. First, once the scale factors are chosen, every addition, subtraction, division, and left-shift operation must be checked to insure that no overflows have occurred. If an overflow is detected, corrective action must be taken to insure that the software will not go into an undefined state and cause erroneous control outputs. When testing the control with the hybrid simulation, this would pose no particular problem. During an engine test, however, this could have disastrous results. In addition to taking corrective action, a latch should be set to indicate where the action was taken. This not only aids the programmer in debugging the control, but also gives warning that a problem exists and that immediate abort action should be taken during an engine test. The second problem that emerges is the loss of precision when dealing with very small numbers. This manifests itself in three subroutines: LQR, integral control, and the FTIT estimator. In each row of the LQR, which is a 5×5 matrix multiplication, five small state deviations times five small gains could produce individually small products. However, the total sum across the row could be quite significant. Therefore the entire 30-bit result of the LQR multiplications must be saved, added as 30-bit double-precision numbers, and then truncated back to 16-bit numbers in the final result. Similar double-precision calculations are performed in the integral control and the FTIT estimator to add accuracy to the calculations.

Use of a Large-Scale Mainframe Time-Sharing System

The multivariable control was programmed on the Lewis IBM 360/67 computer. This was possible because a cross-assembler for the SEL 810B is available on the 360 and the the control's object modules (i.e., assembled source code) could be dumped into the Tektronix peripheral equipment in the hybrid facility (fig. 6). These 25 object modules could then be loaded directly into the SEL's memory.

The use of the time-sharing system provides two advantages. First, it makes available a sophisticated editing package so that the different control program subroutines can share the same common block variable declarations, thereby reducing the possibility of one subroutine overwriting the results of another. This reduces the necessary debugging time associated with the integration of the various control subroutines into a total package. Second, the time-sharing system makes it possible to write Fortran programs that can take the matrix and schedule data directly from the Fortran deck. These programs not only scale and organize the data, but can also put it in integer form, which is then processed directly by the cross-assembler. Doing data processing in this manner simplifies the debugging of a control since it capitalizes on its well-organized structure. A further benefit is that the regulator and integral gains can be changed quickly and are relatively error free.

Control Software Configuration

The final software for the multivariable control exists in two configurations: the hybrid simulator version, and the PSL engine test version. Block diagrams for each of these versions are shown in figures 24 and 25, respectively. Both configurations have the same software to compute the multivariable control algorithm, which has been discussed in detail. In addition, they both contain the same data input/output software to perform man-machine communications and real-time steady-state and transient data sampling. However, the other components of each control are different because of their diverse application environments. The hybrid simulator version needs the computer to do the multivariable control, digital actuator simulations, and real-time data sampling all in 10 msec. This is done in a laboratory environment, where there are no requirements for engine safety. The PSL version must also do the multivariable control calculation and data sampling. In addition, it must check for sensor and actuator failure and be able to make sequencing changes to allow the control to be engaged and disengaged from the actual engine hardware. The following sections discuss the software blocks for each control configuration in some detail. Where appropriate, items that apply to both configurations are discussed together.

Actuator simulations. – Actuator simulations are used in both control configurations. In the hybrid configuration they are used to augment the hardware on the hybrid computers, which do the real-time engine simulation. In the PSL version they are used to do steady-state and transient actuator failure checks. The actuator models that are implemented are the full nonlinear actuator models that were supplied by the engine's manufacturer. Note that to provide satisfactory dynamic simulation of all of the actuators, an actuator update time of 2 msec was necessary. How this update time is accommodated is discussed in the sections on the respective control executives.

Data input/output software. – The data input/output software is a general-purpose program package that is common to both control configurations. This software package runs during the control computer's spare time and facilitates man-machine communications. It allows all of the variables in the control to be referenced by using alphanumeric characters and engineering units. It also permits the printing of tables of steady-state variable values and the dynamic display of variables on chart

recorders at scale expansions of as much as 1028 times normal. It allows the sampling of variables during a transient, the placing of these variables in spare memory, and then the dumping of these time histories of the sampled variables to different peripheral units, such as a floppy disk or cathode ray tube display, in assorted formats. The package also provides various debugging capabilities such as displaying and changing memory locations. Further information on this software package, which is called INFORM, is given in reference 14.

Multivariable control memory requirements. – It is important to note that the size of the control software and its complexity are dictated by constraints on the programmer and the architecture of the machine. The constraints on the programmer in this case are (1) that in addition to the control calculation, the computer must do the actuator simulations, transient data sampling, and sensor and actuator failure checks in the PSL version and (2) that all of these functions must be done in the total update time of 10 to 12 msec. The hardware constraints mentioned before are that all of the calculations must be performed by using fixed-point arithmetic; thus overflow checks must be made. Also, the usable memory in the SEL 810B is limited to 16 384 sixteen-bit words. The overall size of the final code is larger than it would be without these programmer and architectural constraints. The reason is that in-line code must be generated for all repetitive calculations since using loops would cause too much time to be lost in incrementing and checking index variables and branching. Furthermore, not only does making overflow checks on the addition, subtraction, and division operations take time and memory, but also the SEL 810B does not allow direct checking of left-shift overflows in its hardware. Therefore shift overflow macro instructions must be inserted to do these checks. This costs additional time and core. If update time ceased to be a constraint, the core size of the program could be reduced by about 10 percent.

Multivariable Control Executive Routines and Timing

Hybrid simulation version. – The function of the executive in the hybrid simulation version of the multivariable control is simply to schedule control calculations, actuator simulations, and data sampling. Figure 26 shows the multivariable control timing diagram. In this figure is one time slice or one 10-msec update interval. The bottom time line shows what happens during the normal control cycle; and the top time line, what happens in addition when transient data sampling takes place.

In the bottom time line an interrupt is generated at time zero to command the block transfer control (BTC) to begin sampling engine-sensed variables for this control interval. Since there are 20 variables and the speed of the A/D converter is 50 μsec/sample, this process takes 1 msec. Note that since the data are sampled via direct memory access, the sampling is done on a cycle-steal basis from the computer. Therefore during this 1 msec, spare-time events can take place. This is discussed in more detail in this section.

After the variables have been sampled, the control algorithm is calculated. It goes from the 1-msec mark to ?. The reason for the indeterminate control termination time is that various paths through the calculation are possible and thus the calculation length can be different each time through. However, it does have an upper limit of 7.45 msec. At the 7.45-msec mark another interrupt is generated to start computation of the actuator outputs based on the control outputs. However, the actuators are not output directly at the end of the control calculation because a 10-msec actuator computation interval provides insufficient bandwidth for an accurate simulation. To provide sufficient bandwidth, a 2-msec update interval is necessary. The control outputs change only every 10 msec. Therefore, knowing that the actuator input is constant, we can calculate its output for the next 10 msec at the required 2-msec intervals. This results in a 37.5-nsec impulse every 2 msec for the actuator outputs. Note that the 1.45-, 3.45-, 5.45-, and 7.45-msec marks are the actuator outputs from the previous update interval. The 9.45-msec impulse is the first output from the current control update interval. The block marked "calculate actuator simulations" in figure 26 then actually represents going through the actuator loop five times, once for each 2-msec update time, to produce the desired actuator dynamics. Once the actuator calculation has been done, the control waits for the next control sampling interrupt to begin the cycle again.

In addition to the control calculations and actuator simulations, the computer also performs a data-sampling function. This is represented by the top time line in figure 26 and runs asynchronously to the events on the bottom time line. The top time line shows that no matter what the control is doing, if a sampling interrupt comes in, another block transfer control, which is similar to the one used for main control sampling, is sent out to gather data. Once the data have been gathered by the direct memory access controller, the sampled variables are stored away in extra core for later dumping to the floppy disk. Note that the storing process and the dumping of the data to disk takes place during the times shown on the diagram and during the 1-msec main MVC sampling time (i.e., spare time). The capability to sample in an asynchronous manner relative to the control allows the varying of the sample interval. Therefore it is just as easy to take samples at 20, 100, or 500 msec.

Experimental (PSL) Version

The PSL control executive works in basically the same way as the executive for controlling the hybrid simula-

tion. However, there are two notable exceptions. First, the PSL executive must contain mode-switching logic that signals when to track, run, and abort the control. This mode switching is necessary to smoothly operate the control in PSL. Second, the executive need not put out simulated actuator output every 2 msec to preserve the actuator dynamic response. The control operates according to the timing diagram in figure 27. Note that the update interval has grown to 12 msec. This is necessary because in addition to the tasks performed in the hybrid version, checks must be made to detect sensor and actuator failures. At time zero an interrupt is generated to start the sampling of engine sensors. This takes 1 msec. For the next millisecond the sensor failure checks are run. Then, the actual multivariable control algorithm calculation is performed. This takes 6.5 msec. Next the feedback failure checks are performed along with any control sequencing changes. The control outputs are then output to the research actuators and to the actuator simulations. These actuator simulations are used to perform the feedback checks during the next update interval. Since these simulations are the same as the ones used in the hybrid configuration, they must each be computed six times with an effective 2-msec update time to insure that the digital simulations have the proper bandwidth. Spare time for the computer occurs in two places: the time between the finish of the actuator calculation and the occurrence of the timer interrupt signaling the start of the next control calculation, and the time during which the variables are being sampled by the block transfer controller. During these spare times the data input/output software does its man-machine communications and its steady-state or transient data taking and display.

The control mode or control sequencing changes are done to allow for smooth transition from the backup control to the multivariable control and back. The control executive in the PSL version of the control has three modes: setup, run, and scram. In the setup mode the control is run and the outputs are compared with the sensed actuator feedbacks. If all of the outputs are within a specific tolerance of the feedback signal for 5 sec, the control is put into the run mode. In the run mode the control functions are the same as in the setup mode, the only difference being that in the run mode a permit signal is sent to PSL to allow the actuator panels to be switched from the backup control to the multivariable control. If the software failure checks detect a problem, the control switches from the run mode to the abort mode. In the abort mode the control releases the permit signal, and this causes the actuator panels in PSL to revert to the backup control system. In addition, the control software is frozen at that point to allow inspection of the control inputs and outputs at the moment of scram in order to help determine the cause of the abort.

Input Failure Logic

The purpose of the input failure logic is to determine the validity of a particular sensor signal. This is done by using a subset of four possible checks on each signal that is fed to the control. The first check is a minimum-maximum check

$$S_{\min} \leq S_n \leq S_{\max}$$

where

S_n current value of sensed signal
$S_{\min}$ minimum value of sensed signal
$S_{\max}$ maximum value of sensed signal

This check allows the detection of a gross, hard, out-of-range failure that would be erroneous anywhere within the engine's flight envelope. The second check is a DELTA check:

$$|S_n - S_{n-1}| \leq E$$

where

S_{n-1} value of sensed signal for last update interval
E error tolerance within which difference must lie

By comparison of the present and past values of the signal, an erratically responding sensor can be detected and, in addition, a hard sensor failure can be anticipated.

The third check is a percent-of-point-deviation check:

$$S - (N\ \%S) \leq S_n \leq S + (N\ \%S)$$

where

$S \pm (N\ \%S)$ nominal value of sensor plus or minus a certain percentage

This is basically a minimum-maximum check with the error bounds defined as being plus or minus a given percentage of the original sensed value. This check is effective only on variables that are not expected to change with flight conditions. However, with the minimum-maximum error bounds considerably narrower, a hard failure or a slow drift condition can be detected much more quickly. The fourth check is a reference-point deviation check:

$$|S_n - \tilde{S}_{n-1}| \leq E$$

where

$\tilde{S}_{n-1}$ modeled value of sensor from previous update interval

This check uses the control's reference-point schedules and transition control to produce steady-state and transient models for the signal that can be compared with the actual sensed value. This error check not only provides for the detection of hard failures and failures caused by erratically responding sensors, but also allows for the detection of slow drift or a sensor whose dynamic response or time constant may be too low.

Actuator Failure Check

The actuator failure checks use the complete nonlinear actuator simulations for the fuel flow, exhaust nozzle, RCVV, and CIVV actuators to compute a modeled value of the actuator output. These are the same simulations used in the hybrid simulator version of the control software. This modeled value is then used to implement a check in the following manner:

$$|A - \tilde{A}| \leq E$$

where

A sensed value of actuator feedback
$\tilde{A}$ modeled value of actuator feedback

This check allows detection of errors in the forward loop such as D/A converter failures, transmission line failures, actuator panel failures, and actuator hardware failures before they are detected indirectly through the sensors and do inadvertent damage to the engine.

Discussion of Sensor and Actuator Failure Checks

Table IV shows all of the sensed signals, including actuator feedbacks, grouped according to the types of failure checks used on each. In addition, where applicable the error tolerance and other limit values are given for the respective sensors. Note that a minimum-maximum failure check is performed on each signal to provide an initial failure screen for a hard failure. In addition, checks are added to each group to provide additional coverage. The percent-of-point-deviation checks are made on the tunnel conditions (i.e., PT2MES, TT2MES, and P0MES) since they will vary only a small percentage at any given flight condition. Clearly, if this type of system was being used in a flight environment, independent checks of this type (i.e., very tight minimum-maximum limits) could be made by using Mach number and altitude information from the airplane's central air-data computer. The reference-point deviation checks are performed on SN1EST, SN2EST, PT6CEST, PT6HEST, WFMBFM, and PT4EST. These signals are all modeled by the control's reference-point schedules and transition logic to provide open-loop trajectories for the linear quadratic regulator. Therefore the magnitude of LQR state deviations is being examined at all times to determine a sensor failure. Note that PT6CEST and PT6HEST are averaged to provide the control with the final PT6EST signal used by the control. However, observation of the two sensed pressures

TABLE IV. - SENSOR AND ACTUATOR FAILURES

[Values are in percent of full scale.]

Signals	Minimum-maximum checks		DELTA	Percent of point deviation	Reference-point deviation check		Actuator model check	
	Minimum	Maximum			Steady state	Transient	Steady state	Transient
PT2EST	1.2	97		±10				
TT2EST	-2.6	43		±1				
P0MES	0.25	70		±7.5				
SN1EST	20	77			6	1.3		
SN2EST	57	93			5.3	7.3		
PT6CEST	1	65			15	25		
PT6HEST	1	65			15	25		
PT4EST	3.3	67			8.2	14.6		
WFMBFM	0.6	82			9	21	9	12
PT25H	1	90	5					
PT25C	1	90	5					
PS25H	1	90	5					
PS25C	1	90	5					
FTITMES	28	60	2.4					
T25MES	4.5	55	3.4					
PLAMV	6.6	66.6	4.1					
BLEED (AREA)							-10	60
RCVV							18	27
CIVV							24	36
AJ							4.2	9.8

indicates that they are fairly close and can be modeled accurately by the PT6TR value in the control.

The DELTA checks are used to give anticipatory action on signals that have no modeled values and no attributes that would allow limitation of their valid range. However, failure of any one of these signals may not be catastrophic. The undetected failure of the raw signals used to compute the $\Delta P/P$ parameter (PT25H, PT25C, PS25H, and PS25C) would only result in the exhaust nozzle closing a very small amount. For a PLAMV failure the engine would be oscillating and unable to hold a reference point. Failure of FTITMES would not be desirable since it could result in a possible overtemperature if the engine was being run at military power. A T25MES failure could result in an engine stall because of compressor geometry misscheduling. But since both sensors are thermocouples, they will most likely fail in a hard catastrophic mode, which is easy to detect.

The actuator model checks are performed on the WFMBFM, AJ, CIVV, and RCVV feedback signals. The bleed feedback is given only a minimum-maximum check because it is closed except during a throttle chop.

For all of the failure checks discussed, a signal has to fail four consecutive times to be declared bad. Although 10-Hz analog filters are used on each signal, this condition was added to provide extra protection against spurious noise coupling inadvertently scramming the control.

The error tolerance and limit values for all of the failure checks that are summarized in table IV were obtained from the hybrid simulator evaluation results except the raw $\Delta P/P$ parameter and error tolerances for the actuator model checks. The others were obtained by running transients and by some trial and error using the hybrid simulation. The reference-point deviation checks and the actuator model checks have two sets of error tolerances: one for steady state, and one for transient. Obviously this is desirable since most of the test time is spent at steady state and the dynamic models are not as good as the ones for steady state. Therefore, to provide rapid detection of failures 99 percent of the time, one set of error tolerances is used. These error tolerances are increased during a transient to prevent detection of false failures. A transient is the time that the MTRAN flag is equal to 1 until 1 sec after the flag returns to zero.

Output Processing and Failure Checks

Output processing and failure checking scales the control outputs to make them compatible with the inputs to the research actuators. In addition, it performs the following checks on fuel flow, exhaust nozzle area, rear compressor variable vanes, and compressor-inlet variable vanes:

$$|O_n - O_{n-1}| \le E$$

where

O_n current value of control output

O_{n-1} past value of control output

This software allows one last check on the control's health by making sure the outputs are not behaving erratically. This erratic behavior could be caused by an undetected overflow in an arithmetic or shift operation or possibly by an actual hard failure of the computer arithmetic unit. The error tolerances in this application were derived by analyzing hybrid data; however, it is necessary to fail this check only once to be considered a failure. The reason is that failure here would indicate a possible catastrophic computer problem. Scramming to backup control should be done as quickly as possible. Furthermore, because of the control's filtering action, noise on the input sensors is not a problem with this check.

Multivariable Control Memory Requirements

The memory requirements for the hybrid and PSL versions of the multivariable control are given in tables V and VI, respectively. The tables give a more detailed breakdown of each version of the control than the figures. In particular the multivariable control algorithm is broken down into each of the component parts that are described in detail in the preceding section. Included in the total are the subroutines that the control uses to generate nonlinear functions. The block data consist of the nonlinear schedular data and the control matrix data.

TABLE V. - CORE REQUIREMENTS FOR MVC PROGRAM (HYBRID)

Multivariable control algorithm:		
FTIT estimator	309	
Reference-point schedules	618	
Gain control	834	
Transition control	632	
Integral control	783	
LQR control	347	
Engine protection	198	
Function generation	370	
Total		4091
Block data:		
Schedules	1226	
Matrices	736	
Total		1962
Actuator simulations		232
Control executive		860
Grand total		7145
General-purpose input/output and debug		5694

TABLE VI. - CORE REQUIREMENTS FOR MVC PROGRAM (PSL)

Multivariable control algorithm:		
FTIT estimator	309	
Reference-point schedules	618	
Gain control	834	
Transition control	632	
Integral control	783	
LQR control	347	
Engine protection	198	
Function generation	370	
Total		4091
Block data:		
Schedules	1752	
Matrices	736	
Total		2488
Sensor checks		1169
Actuator simulations and checks		381
Output checks		193
Control executive		1208
Grand total		9530
General-purpose input/output and debug		5694

There are more schedular data for the PSL version than for the hybrid version because nonlinear actuator conversion curves and thermocouple curves are added. The addition of sensor, actuator, and output checks, along with a more sophisticated executive to allow for easy control engagement, makes the PSL version approximately 2.5 kilowords larger than the hybrid version. For each configuration the general-purpose input-output debug package has been omitted from the computation of the grand total because this software would not be necessary or included in an embedded control system on an engine coming out of the factory. Although the software is invaluable in the research environment in which it was run, including it in program size totals gives the impression of more control system complexity than exists.

Results and Discussion

One of the major results of this program has been to demonstrate that it is possible to design a multivariable control for a state-of-the-art turbofan engine. In addition, it is possible to implement this control, along with sensor and actuator failure logic, on a minicomputer characteristic of flight-qualified hardware available today and meet the control timing and computer memory size requirements.

Since this control is believed to be characteristic of the ones to be used on future engines, observations can be made that are pertinent to future controls development. The successful implementation required almost 80 percent of the computer's resources (fig. 27 and table VI). To achieve this, it was necessary to program the control in assembly language and to use fixed-point arithmetic. Although fixed-point arithmetic is fast on this computer since it has the proper hardware (table I), it is necessary to work always with scaled numbers and to make overflow checks on all addition, substraction, and division operations. This is necessary to insure that, if an overflow occurs, it will be detected and a fail-safe condition will occur.

What is disturbing about this is that future engine controls may contain not only multivariable control algorithms but also sensor failure accommodation control, performance optimization, and self-trimming. These are all complex algorithms employing a large amount of arithmetic computation. This is contrasted to the simulation of the relatively simple hydromechanical control, which does mostly table lookups and makes logical decisions (ref. 15). Experience in the computer industry and in the use of computers by the military has shown that when hardware is used more than 50 percent of the time, the software programming time increases exponentially (ref. 16). In addition, by 1985, software costs will account for 90 percent of the cost of a computer system (ref. 16).

Therefore, because the software cost and complexity of engine controls will continually be increasing, the use of more advanced hardware and software concepts must be considered in the design of engine controls. Specifically, these should include the use of computer architectures that employ hardware floating-point processors and the programming of controls by using sophisticated optimization compilers that allow structured programming. The floating-point processors will liberate the programmer from the tedium of using scaled numbers. Furthermore the use of structured programming will not only decrease the costs of initial program development, but also simplify debugging and the task of software maintenance. Use of these items will definitely decrease future controller overall costs.

National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio, February 28, 1983

Appendix—Symbols

AFB	FTIT estimator feedback gain constant
AJ	exhaust nozzle position feedback
AJACH	scheduled value of exhaust nozzle position
AJCL	unlimited exhaust nozzle position command
AJCOM	limited final value of exhaust nozzle position command
AJDEL	jump value of transition exhaust nozzle area
AJMN	value of minimum allowed exhaust nozzle position command
AJMX	value of maximum allowed exhaust nozzle position command
AJPARM	parameter used to schedule minimum and maximum exhaust nozzle position commands
AJQI	LQR contribution to exhaust nozzle position command
AJRATE	value of exhaust nozzle transition rate
AJTR	transition value of exhaust nozzle position
ALF	fuel feedforward gain constant in FTIT estimator
AMNCV	schedule to determine minimum exhaust nozzle position command
AMXCV	schedule to determine maximum exhaust nozzle position command
BLC	feedback value of bleed position
BLCI	LQR contribution to final bleed command
BLCCL	unlimited bleed position command
BLCCM	limited final value of bleed command
BLCMN	value of minimum bleed command
BLCMX	value of maximum bleed command
BLCTR	transition value of bleed
BLEED	feedback position of bleed valve
CFXX	general designation for final interpolated gain matrix values
CHXX	general designation for interpolated value of high-power gain matrices
CIVV	compressor-inlet variable vane position feedback
CIVVCL	unlimited compressor-inlet variable vane position command
CIVVCM	limited final value of compressor-inlet variable vane position command
CIVVMN	minimum scheduled value of compressor-inlet variable vane positions
CIVVSH	scheduled value of compressor-inlet variable vane position
CLQI	LQR contribution to compressor-inlet variable vane position command
CLXX	general description of interpolated value of low-power gain matrices
COMCOR	sensed value of compressor speed corrected to station 2.5
COMRAT	value of compressor speed transition rate
CVMXCV	flutter boundary schedule
CWF	raw value of scheduled corrected fuel flow
CWFCRV	corrected fuel flow schedule
CWFMB	CWF times WFROT
CX1C . . . CX1F CX2C . . . CX2F	interpolated gain matrix values
CX28	interpolated gain matrix value
DELTM	intermediate value in estimator equal to FTITMES – FTITSH
DELTA	square root of PT2MES divided by 14.7
DELTH	sum of XI FTIT estimator, state, and feedforward fuel flow component
DELTRN	generalized interpolation logic block for all transition rates
DEN	density parameter used for gain scheduling

DENLIMIT	low-power DEN parameter limiting curve (fig. 15)
DIFF	difference between scheduled and transition values of compressor speed
DIR	difference between limited and unlimited fuel flow commands
DN1	low rotor speed state deviation
DN2	high rotor speed state deviation
DP25AD	ΔP schedule modifier for high inlet temperatures
DP25CRV	raw $\Delta P/P$ schedule
DP25DCRV	modifier curve for $\Delta P/P$ schedule
DP25E	sensed value of $\Delta P/P$
DP25SH	final scheduled value of $\Delta P/P$
DP25SI	raw $\Delta P/P$ scheduled value
DP25SN	sensed value of $\Delta P/P$
DPB	combustor pressure state deviation
DPT6	afterburner manifold pressure state deviation
DWF	fuel flow state deviation
EBLC	bleed command integrator error
ECIVV	CIVV command integrator error
EDP25	$\Delta P/P$ integrator error
EFTIT	fan turbine-inlet temperature integrator error
EPBMN	minimum combustor pressure integrator error
EPBMX	maximum combustor pressure integrator error
ERCVV	rear compressor variable vane command integrator error
ESN1	low rotor speed integrator error
E1,E2,E3	state deviation errors in fan turbine-inlet temperature estimator
F	output of PDIFF schedule in transition control jump logic
FANCOR	low rotor speed corrected to total temperature at station 2
FANMAX	schedule to limit maximum scheduled fan speed
FANRAT	value of fan speed transition rate
FDWF	final fuel flow feedforward contribution to estimated fan turbine-inlet temperature
FFWD	scaled value of difference between commanded fuel flow and transition fuel flow
FG	product of values F and G
FMAX	upper Y breakpoint of PDIFF curve
FMIN	lower breakpoint of PDIFF curve
FMPY	scheduled FTIT offset correction for PSL
FMUL	multiplier factor to determine weighting of high- and low-power matrices in final gain matrix
FTICRV	maximum fan turbine-inlet temperature schedule
FTIRAT	transition rate for fan turbine-inlet temperature
FTITEST	estimated value of fan turbine-inlet temperature
FTITMES	measured value of fan turbine-inlet temperature
FTITMX	maximum scheduled value of fan turbine-inlet temperature
FTITSH	scheduled value of fan turbine-inlet temperature
FTITTR	transition value of fan turbine-inlet temperature
FTMXB	bias factor for FTICRV curve
FTWF	scaling parameter output by CFTWTCV curve
FTWTCV	fuel flow scaling parameter schedule used in fan turbine-inlet temperature estimator
G	transition control jump logic parameter
GPARM	limited value of DEN from DENLIMIT schedule
GVIPOS	fan inlet guide vane position
HG1 . . . HG4	constants indicating DEN breakpoint between four high-power gain matrices
HMUL	interpolation factor used in computing final high-power gains
HVSPOS	rear compressor variable vane position
IBLC	bleed integrator output
ICIVV	compressor-inlet variable vane integrator output

IDP25	exhaust nozzle integrator output
IRCVV	rear compressor variable vane integrator output
IWFMB	fuel flow integrator output
LG1,LG2	constants indicating DEN breakpoints between two low-power gain matrices
LMT5	low rotor speed integrator trim flag
LMT6	fan turbine-inlet temperature integrator trim flag
LMT7	high combustor pressure integrator trim flag
LMT8	low combustor pressure integrator trim flag
LMUL	interpolation factor used in computing final low-power gains
MAJFLG	exhaust nozzle integrator inhibit flag
MBLFLG	bleed integrator inhibit flag
MCVFLG	compressor-inlet variable vane inhibit flag
MPBHI	high combustor pressure limit flag
MPBLO	low combustor pressure limit flag
MRCFLG	rear compressor variable vane integrator inhibit flag
MTFFLG	fan turbine-inlet temperature limit flag
MTRAN	barge transition indicator flag
MWFFLG	fuel flow integrator inhibit flag
NLBIAS	low rotor speed bias factor to match hybrid model to actual engine
N1C2	input parameter of flutter boundary schedule CVMXCU
N1P2BI	bias factor for compressor-inlet variable vane flutter boundary calculation
N1P2CV	schedule that computes bias factor for compressor-inlet variable vane flutter boundary calculation
N2HICV	curve to compute intermediate breakpoint for high rotor speed and transition rate scheduling spline
N2LOCV	curve to compute low breakpoint for high rotor speed and transition rate scheduling spline
N2MICV	curve to compute intermediate breakpoint for high rotor speed and transition rate scheduling spline
N2XICV	curve to compute high breakpoint for high rotor speed and transition rate scheduling spline
PAR1CV	curve to compute DEN modifying factor for transition and jump rates
PAR2CV	curve to computer DEN modifying factor for transition and jump rates
PARM1	DEN modifying factor for transition and jump rates
PARM2	DEN modifying factor for transition and jump rates
PBBIAS	combustor pressure bias factor to match hybrid model to actual engine
PBCRV	curve to compute schedule PT4MES/PT2MES
PB1CRV	curve to backcalculate SNCCR2
PBMAX	maximum scheduled value of combustor pressure
PBMIN	minimum scheduled value of combustor pressure
PBOPT2	minimum value of combustor pressure divided by inlet total pressure
PDIFF	schedule to limit PPDF to between zero and one
PLAD	past value of limited PLAMV
PLALMT	schedule to limited PLAMV
PLAMP	limited value of PLAMV
PLARAT	schedule to limit rate of change of PLADIF
PLAMV	sensed value of power lever angle
POMES	sensed inlet static pressure
PS25C	static duct pressure at station 2.5
PS25H	static core pressure at station 2.5
PT25C	total duct pressure at station 2.5
PT25H	total core pressure at station 2.5
PLADIF	rate of change of PLAMV at high power
PPDF	rate-limited change of PLAMV at high power
PT2MES	sensed value of inlet total pressure at station 2
PT4EST	sensed value of combustor pressure

PT4RAT	combustor pressure transition rate
PT4SCH	limited scheduled value of combustor pressure
PT4TR	transition value of combustor pressure
PT4SCI	unlimited scheduled value of combustor pressure
PT6CRV	curve that computes scheduled value of PT6
PT6CEST	sensed value of duct total pressure at station 6
PT6EST	sensed value of afterburner manifold pressure
PT6HEST	sensed value of core total pressure at station 6
PT6RAT	afterburner manifold pressure transition rate
PT6SCH	biased limited scheduled value of afterburner manifold pressure
PT6SCI	raw scheduled value of afterburner manifold pressure
PT6TR	transition value of afterburner manifold pressure
P4OPT2	scheduled value of combustor pressure divided by inlet total pressure
P6BIAS	bias factor for scheduled afterburner manifold pressure
RATCV	series of schedules to rate limit changes in transition values
RCMNCR	schedule to compute minimum allowable rear compressor variable vane command
RCMXCR	schedule to compute maximum allowable rear compressor variable vane command
RCVCRV	schedule to compute steady-state value of rear compressor variable vane command
RCVV	sensed rear compressor variable vane position
RCVVCL	unlimited rear compressor variable vane command
RCVVCM	limited rear compressor variable vane command
RCVVMN	minimum allowable rear compressor variable vane command
RCVVMX	maximum allowable rear compressor variable vane command
RCVVSH	scheduled value of rear compressor variable vane angle
RLQI	LQR contribution to rear compressor variable vane command
SMNCRV	curve that computes maximum scheduled high rotor speed
SMNMES	sensed propulsion system Mach number
SNCCR2	scheduled corrected high rotor speed
SNCMAX	maximum scheduled high rotor speed
SNCSCH	limited scheduled high rotor speed
SNCSCI	unlimited scheduled high rotor speed
SNCTR	transition value of high rotor speed
SNFCRV	curve that computes scheduled high rotor speed
SNFMAX	maximum value of scheduled fan speed
SNFSCH	limited scheduled fan speed
SNFSCI	unlimited scheduled fan speed
SNFSCR	corrected scheduled fan speed
SNFTR	transition value of fan speed
SN1EST	sensed value of fan speed
SN2EST	sensed value of high rotor speed
SN2HI	high breakpoint for high rotor speed and transition rate scheduling spline
SN2INT	intermediate breakpoint for high rotor speed and transition rate scheduling spline
SN2LOW	low breakpoint for high rotor speed and transition rate scheduling spline
SN2MID	intermediate breakpoint for high rotor speed and transition rate scheduling spline
S2	switch to uptrim scheduled combustor pressure
S4	switch to freeze all transition control rates
THETA	square root of TT2 divided by 459.7
TITCRV	curve to compute scheduled fan turbine-inlet temperature
TITOT2	scheduled value of FTIT divided by TT2
TT2MES	measured value of total temperature at station 2
T25MES	measured value of duct stream total temperature at station 2.5
WACCRV	curve to compute scheduled airflow
WACHI	curve to compute maximum scheduled airflow

WACLO	curve to compute minimum scheduled airflow
WACMIN	curve to compute minimum overall airflow
WAMNCV	low-power airflow roll-off curve
WDOTS	raw value of scheduled airflow
WDOTSH	final limited value of scheduled airflow
WDOTSI	raw value of scheduled airflow modified by Mach number
WFCOM	limited final value of fuel flow command
WFDEL	step change value for transition fuel flow
WFMAX	maximum value of commanded fuel flow
WFMBCL	unlimited final value of fuel flow command
WFMBFM	sensed value of fuel flow
WFMBSH	limited scheduled value of fuel flow
WFMBTR	transition value of fuel flow
WFMIN	minimum scheduled and commanded value of fuel flow
WFMNN	absolute minimum value for commanded fuel flow
WFOPBL	minimum ratio of scheduled fuel flow to combustor pressure
WFOX	maximum commanded fuel flow
WF/PB	ratio of fuel flow to combustor pressure
WFPBCRV	curve to compute ratio of maximum fuel flow to combustor pressure
WFPBMN	minimum ratio of commanded fuel flow to combustor pressure
WFPBMX	maximum ratio of commanded fuel flow to combustor pressure
WFRATE	transition rate for fuel flow
WFROT	rotation factor for CWFCRV curve
WFSCI	unlimited scheduled fuel flow
WLQI	LQR contribution to final fuel flow command
WOOFBI	bias factor for WFPBCRV curve
XN2HB	high breakpoint used for final gain scheduling
XN2HCV	curve used to determine high breakpoint in final gain scheduling
XN2LB	low breakpoint used in final gain scheduling
XN2LCV	curve used to determine low breakpoint in final gain scheduling
X1,X2,X3	fan turbine-inlet temperature estimator states
Z1,Z2	fan turbine-inlet temperature estimator inputs

References

1. Beattie, E. C.: Control Mode Studies for Advanced Variable-Geometry Turbine Engines. PWA-5161, AFAPL-TR-75-7, Pratt & Whitney Aircraft, 1974. (AD-A009169.)
2. Kwakernaak, H.; and Sivan, R.: Linear Optimal Control Systems. John Wiley & Sons, Inc., 1972.
3. Michael, G. J.; and Farrar, F. A.: An Analytical Method for the Synthesis of Nonlinear Multivariable Feedback Control. UARL-M941338-2, United Aircraft Corp., 1973. (AD-762797.)
4. Bowles, R. J.: Sub-Optimal Control of a Gas Turbine Engine. MS Thesis, Air Force Institute of Technology, Wright Patterson Air Force Base, 1973. (AD-777852.)
5. Michael, G. J.; and Farrar, F. A.: Development of Optimal Control Modes for Advanced Technology Propulsion Systems. UARL-N911620-2, United Aircraft Corp., 1974. (AD-775337.)
6. Stone, C. R.; et al.: Turbine Engine Control Synthesis, Vol. 1. F0164-FR-Vol.-1, AFAPL-TR-75-14-Vol.-1, Honeywell, Inc., 1975. (AD-AD14229.)
7. Weinberg, M. S.: A Multivariable Control for the F100 Engine Operating at Sea-Level Static. ASD-TR-75-28, Aeronautical System Division, Wright Patterson Air Force Base, 1975. (AD-A022699.)
8. Miller, R. J.; and Hackney, R. D.: F100 Multivariable Control System Engine Models/Design Criteria. FR-7809, AFAPL-TR-76-74, Pratt & Whitney Aircraft Group, 1976. (AD-A033532.)
9. DeHoff, R. J.; et al.: F100 Multivariable Control Synthesis Program, Vols. 1 and 2. AFAPL-TR-77-85 Volumes 1 and 2, Systems Control, Inc., 1977. (AD-A052346.)
10. Szuch, J. R.; and Seldner, K.: Real-Time Simulation of the F100-PW-100 Turbofan Engine Using the Hybrid Computer. NASA TM X-3261, 1975.
11. Szuch, J. R.; et al.: F100 Multivariable Control Synthesis Program—Evaluation of a Multivariable Control Using a Real-Time Engine Simulation. NASA TP-1056, 1977.
12. Lehtinen, F. K. Bruce; Costakis, William; Soeder, James F.; and Kurt Seldner: F100 Multivariable Control Synthesis Program—Results of Altitude Tests. NASA TM S-83367, 1983.
13. Arpasi, D. J.; Zeller, J. R.; and Batterton, P. G.: A General Purpose Digital System for On-line Control of Airbreathing Propulsion Systems. NASA TM X-2168, 1971.
14. Cwynar, D. S.: INFORM—An Interactive Data Collection and Display Program with Debugging Capability. NASA TP-1424, 1980.
15. Cwynar, D. S.; and Batterton, P. G.: Digital Implementation of the TF30-P-3 Turbofan Engine Control. NASA TM X-3105, 1975.
16. Boehm, B. W.: Software and Its Impact: A Qualitative Assessment. Tutorial on Software Design Techniques, P. Freeman and A. Wasserman, eds., IEEE, 1977, pp. 12-23.

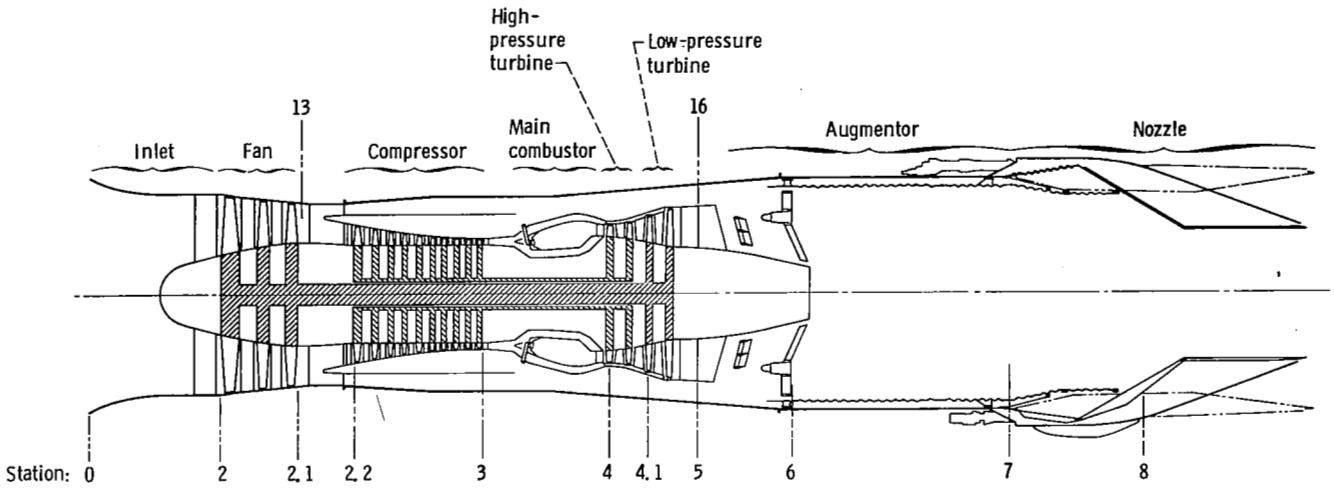


Figure 1. - Schematic representation of F100-PW-100(3) augmented turbofan engine.

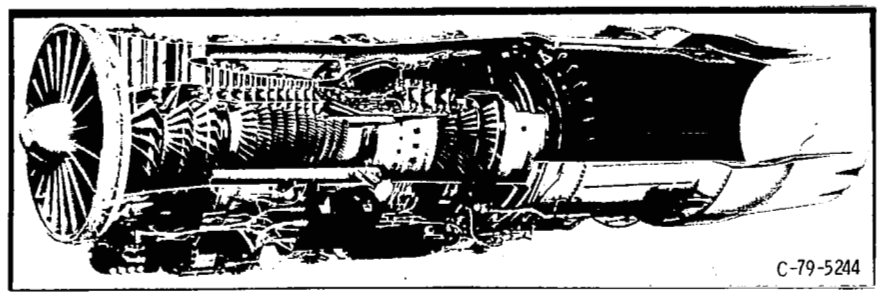


Figure 2. - Cutaway drawing of F100 turbofan engine.

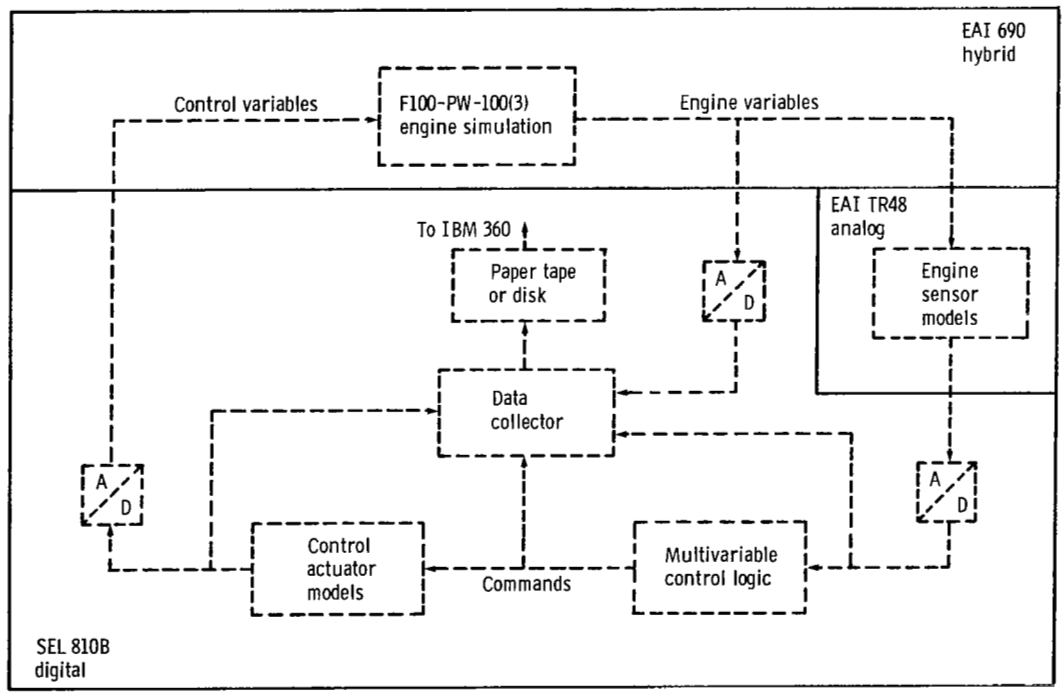


Figure 3. - Schematic representation of hybrid multivariable control evaluation.

Figure 4. - NASA hybrid computing system for real-time engine simulation.

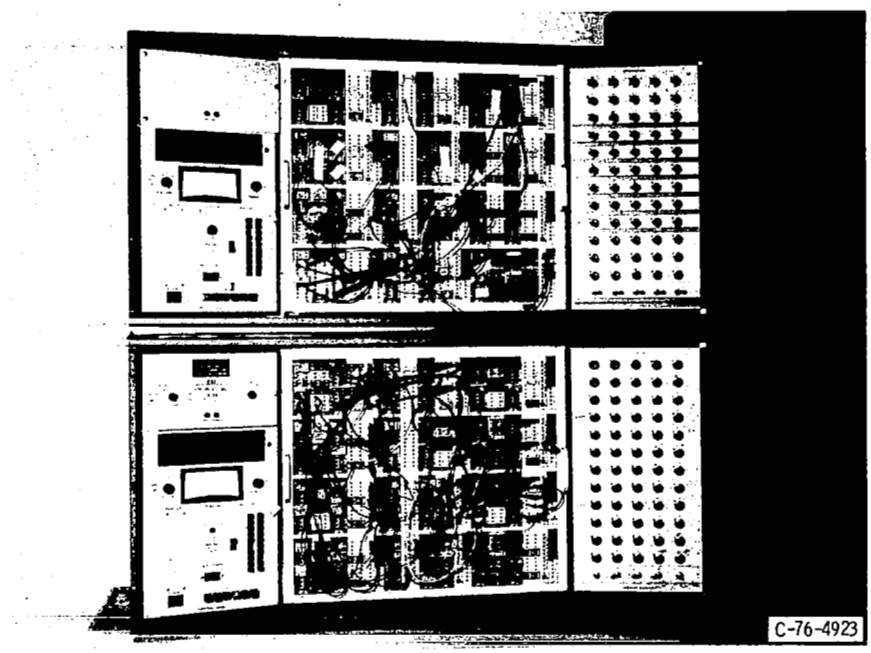


Figure 5. - NASA analog computers used for sensor simulations.

Figure 6. - NASA digital computer input/output and peripheral system.

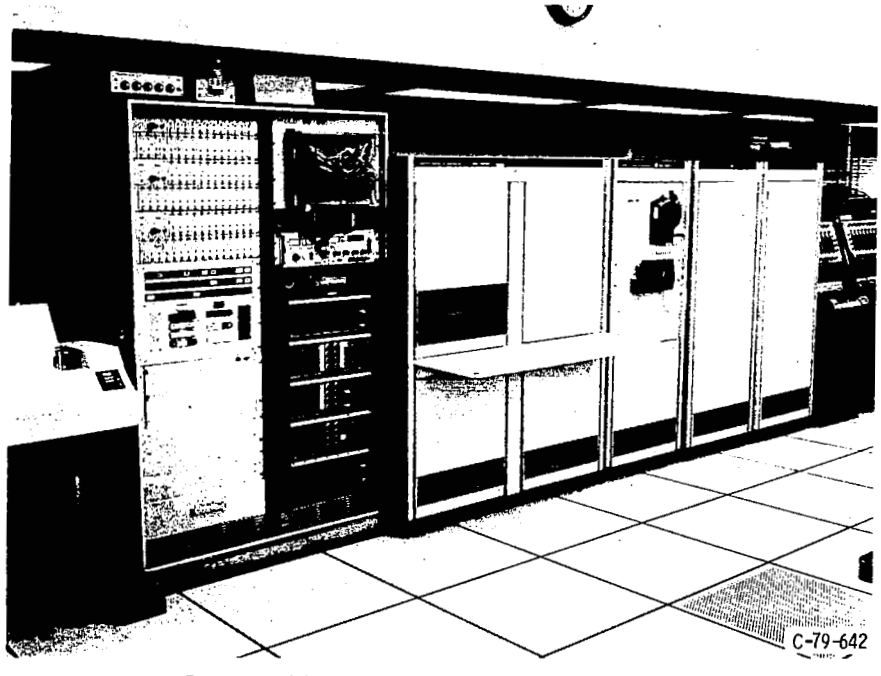


Figure 7. - NASA digital computer system used for on-line engine control.

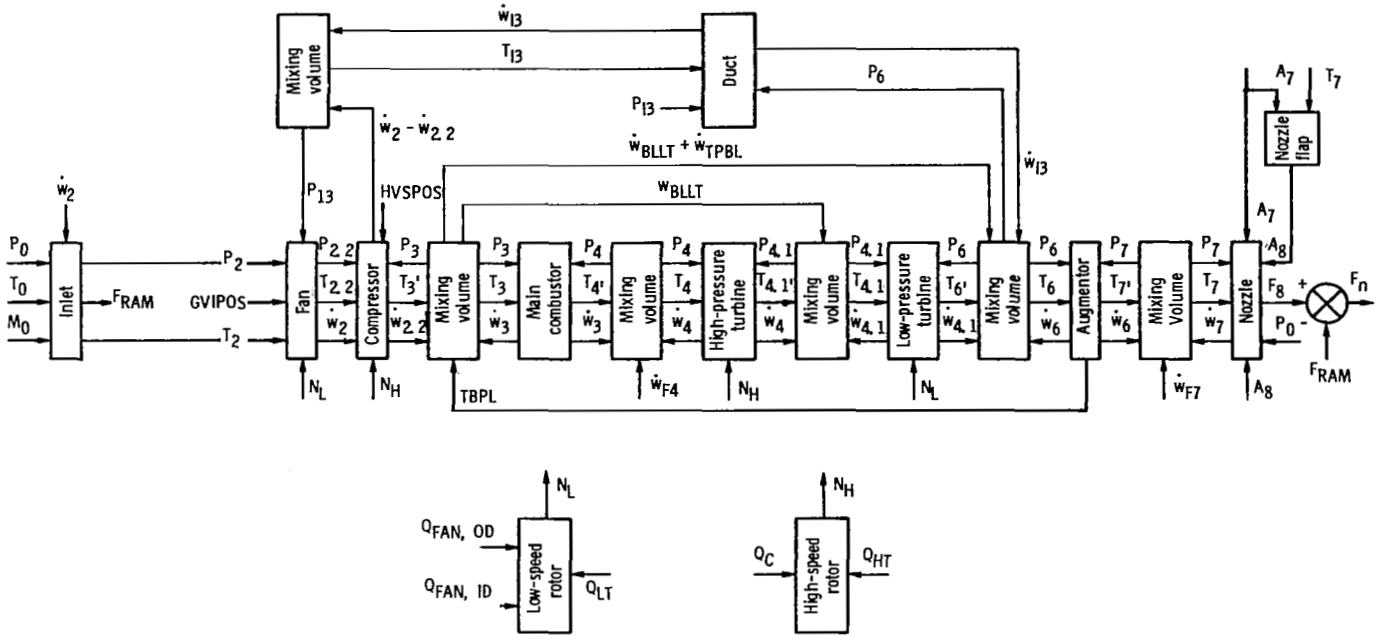


Figure 8. - Computational flow diagram of real-time F100-PW-100(3) engine simulation.

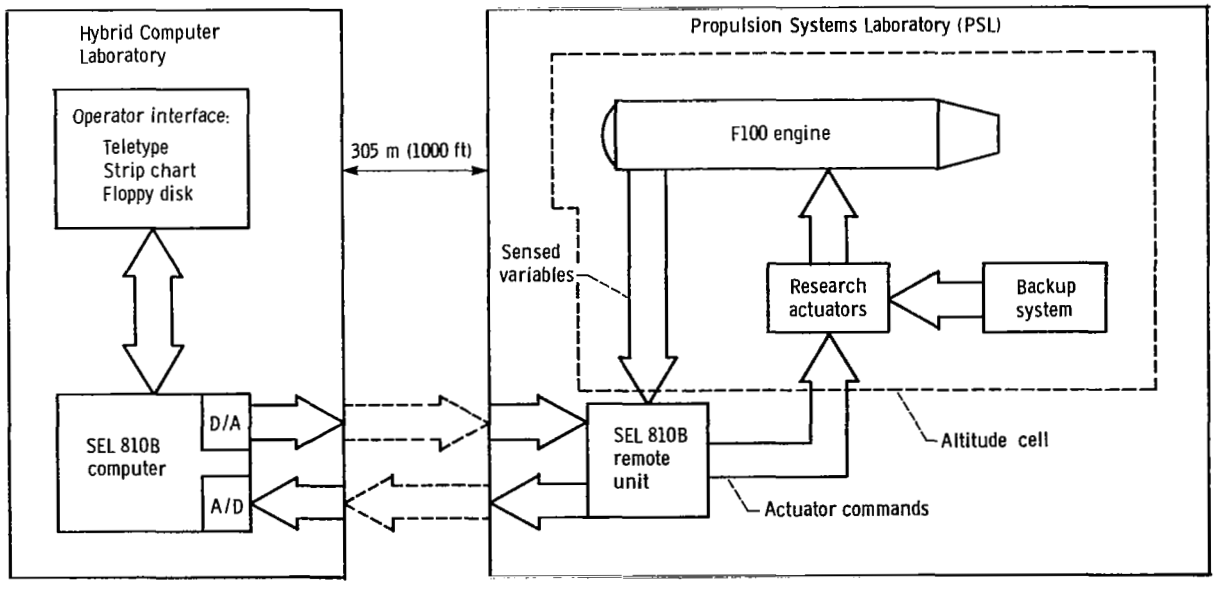
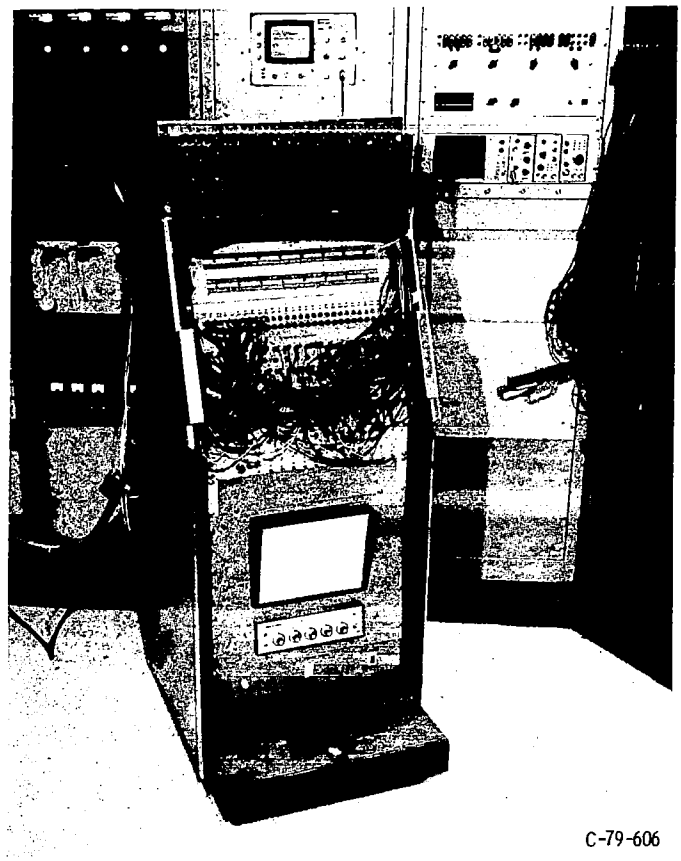
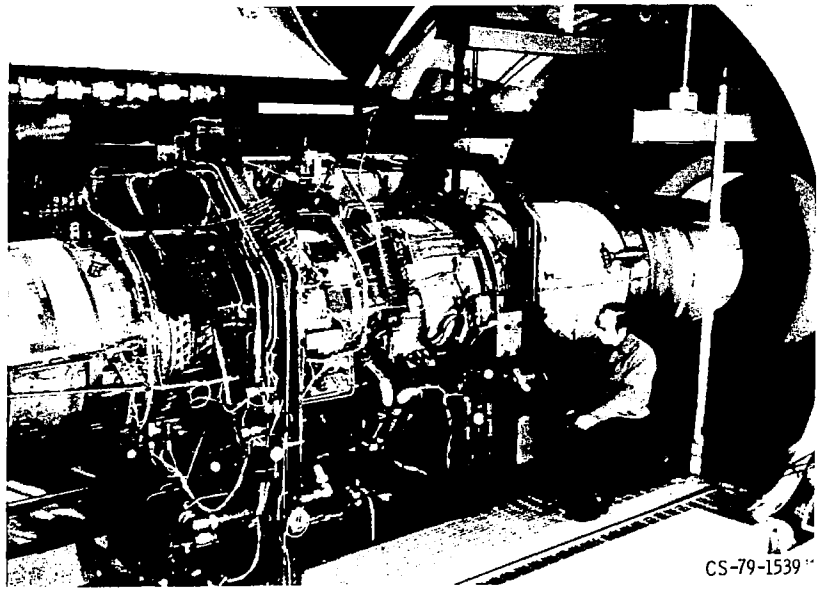


Figure 9. - Schematic representation of PSL multivariable control test setup.



C-79-606

Figure 10. - SEL 810B remote unit.



CS-79-1539

Figure 11. - F100 engine in PSL altitude chamber.

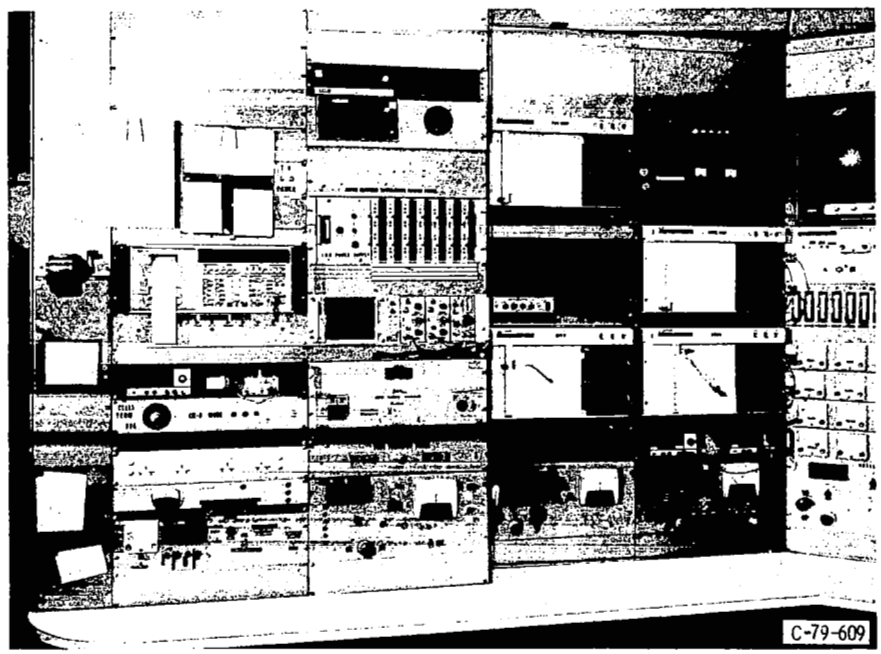


Figure 12. - Research actuator control panels in PSL.

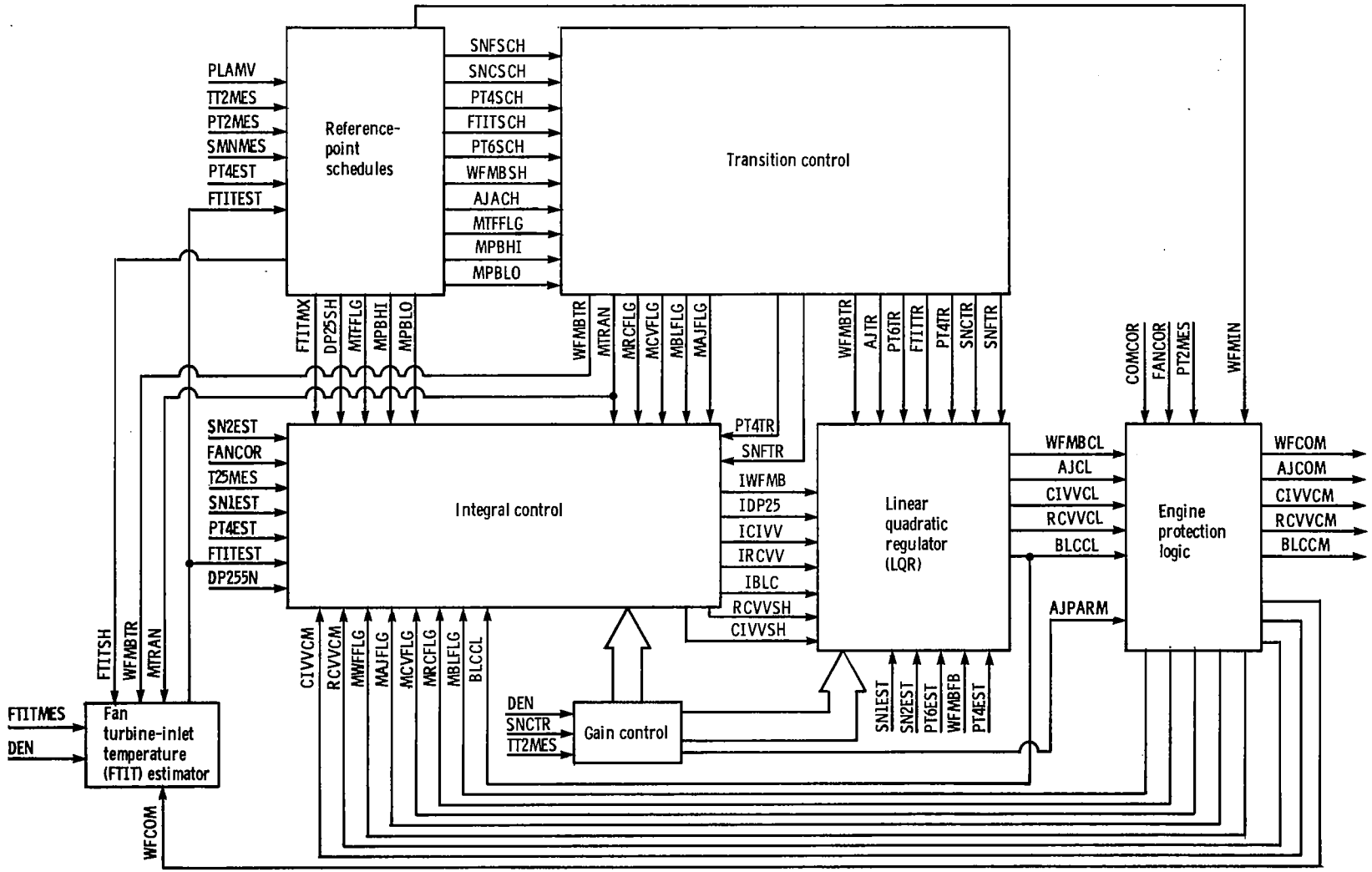


Figure 13. - Block diagram of overall multivariable control.

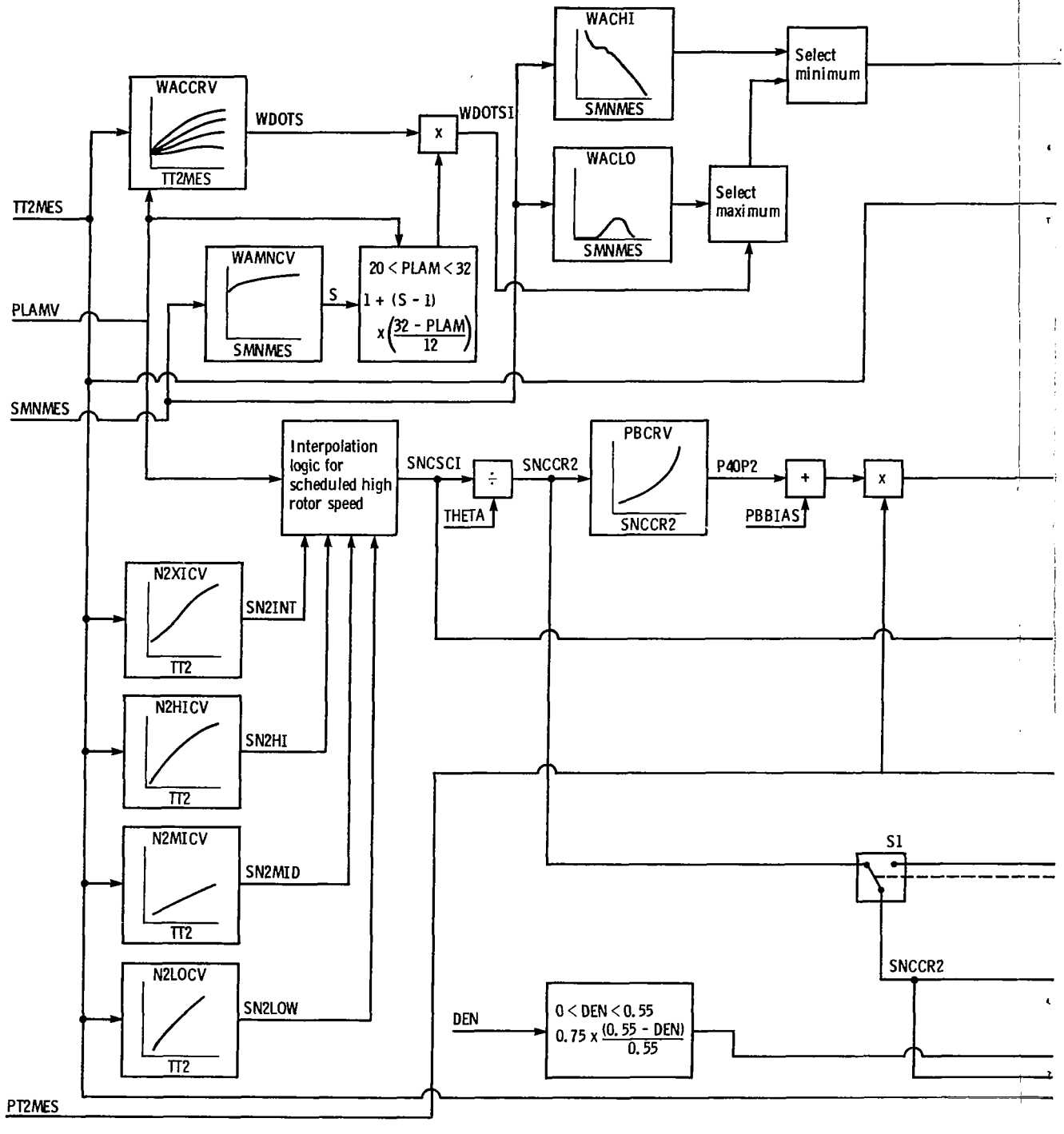
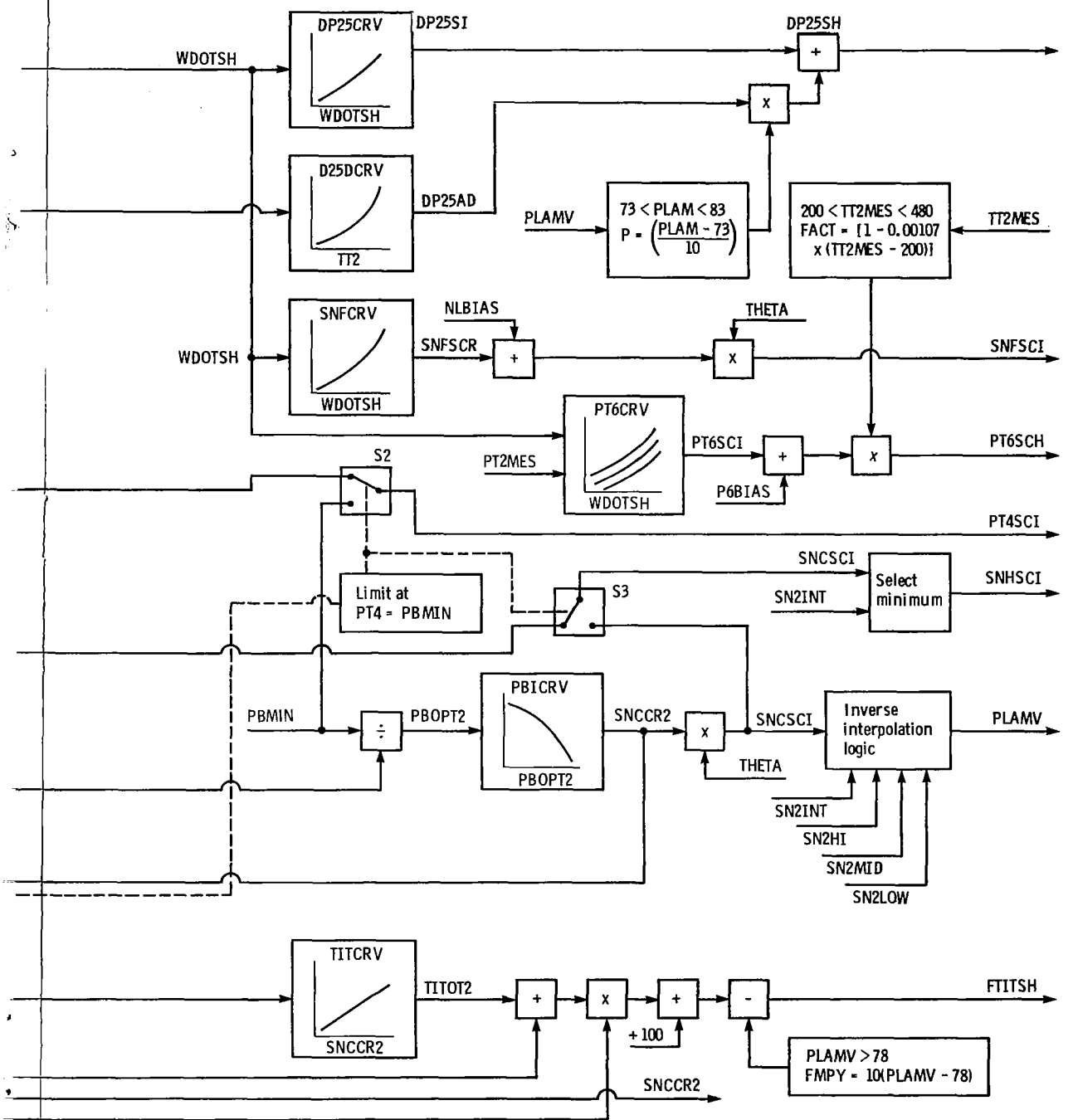


Figure 14. - Multivariable control -



reference-point schedules.

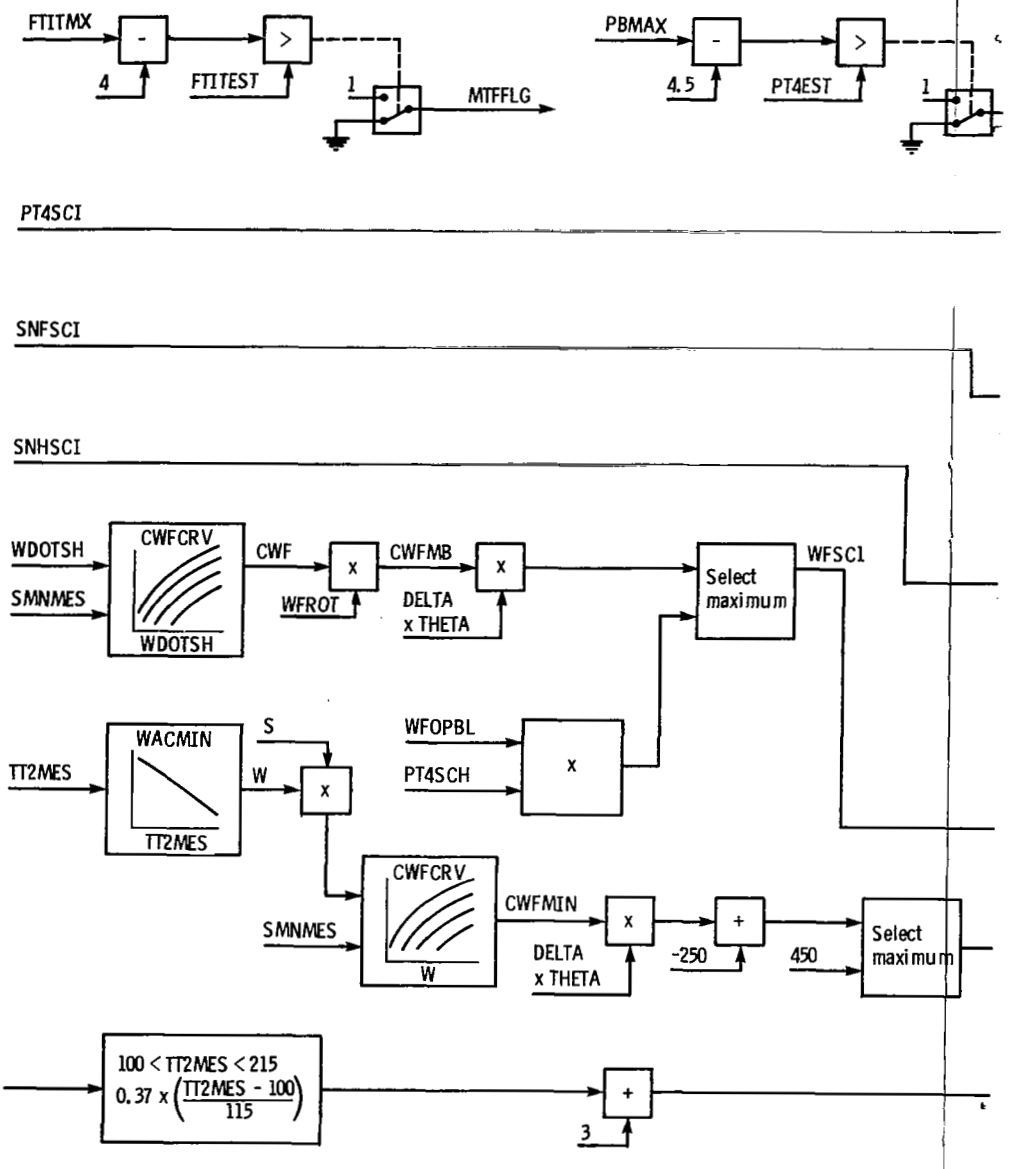
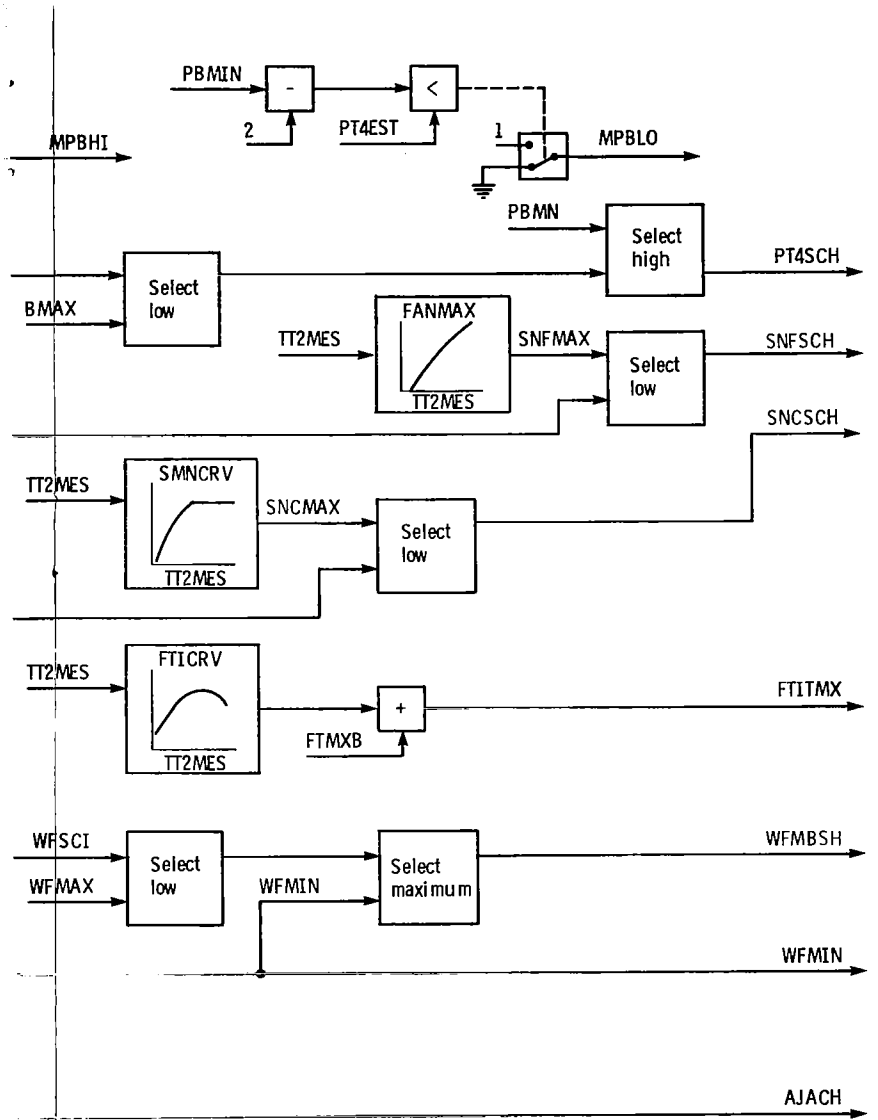

FTITMX
-
4
>
FTITEST
1
MTFFLG
PBMAX
-
4.5
>
PT4EST
1
PT4SCI
SNFSCI
SNHSCI
WDOTSH
SMNMES
CWFCRV
WDOTSH
CWF
x
WFROT
CWFMB
x
DELTA
x THETA
Select
maximum
WFSC1
TT2MES
WACMIN
TT2MES
S
W
x
WFOPBL
PT4SCH
x
SMNMES
CWFCRV
W
CWFMIN
x
DELTA
x THETA
+
-250
450
Select
maximum
100 < TT2MES < 215
0.37 x ((TT2MES - 100)/115)
+
3

Figure 14.


PBMIN
-
<
2
PT4EST
1
MPBHI
MPBLO
PBMN
Select high
PT4SCH
Select low
BMAX
TT2MES
FANMAX
TT2MES
SNFMAX
Select low
SNFSCH
SNCSCH
TT2MES
SMNCRV
TT2MES
SNCMAX
Select low
TT2MES
FTICRV
TT2MES
+
FTMXB
FTITMX
WFSCI
WFMAX
Select low
WFMIN
Select maximum
WFMBSH
WFMIN
AJACH

Concluded.

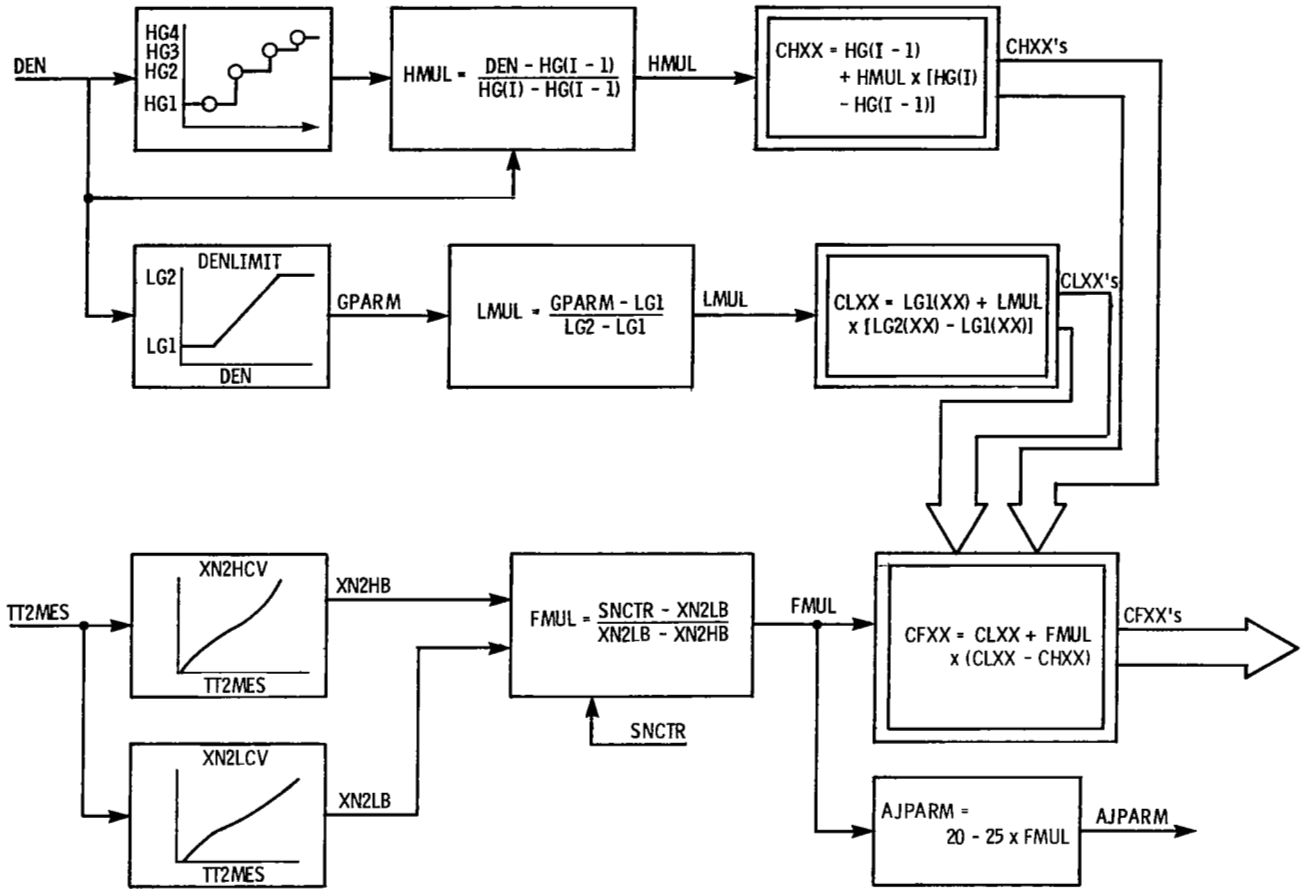


Figure 15. - Multivariable control - gain control.

		State deviations					Spares	Integral output deviations							
		Fan speed	Compressor speed	Afterburner pressure	Fuel flow	Combustor pressure		ΔP/P parameter	CIVV	RCVV	Bleed flow	Fan speed	FTIT maximum limit	Combustor pressure minimum limit	Combustor pressure maximum limit
Controls	Fuel flow	X	X	X	X	X		X	0	0	0	X	X	X	X
	Nozzle area	X	X	X	0	0		X	0	0	0	X	X	X	X
	CIVV	X	0	X	0	0		X	X	0	0	X	0	0	0
	RCVV	X	X	X	0	0		0	0	X	0	0	0	0	0
	Bleed flow	X	X	0	0	X		0	0	0	X	0	0	0	0

Figure 16. - Gain matrix structure.

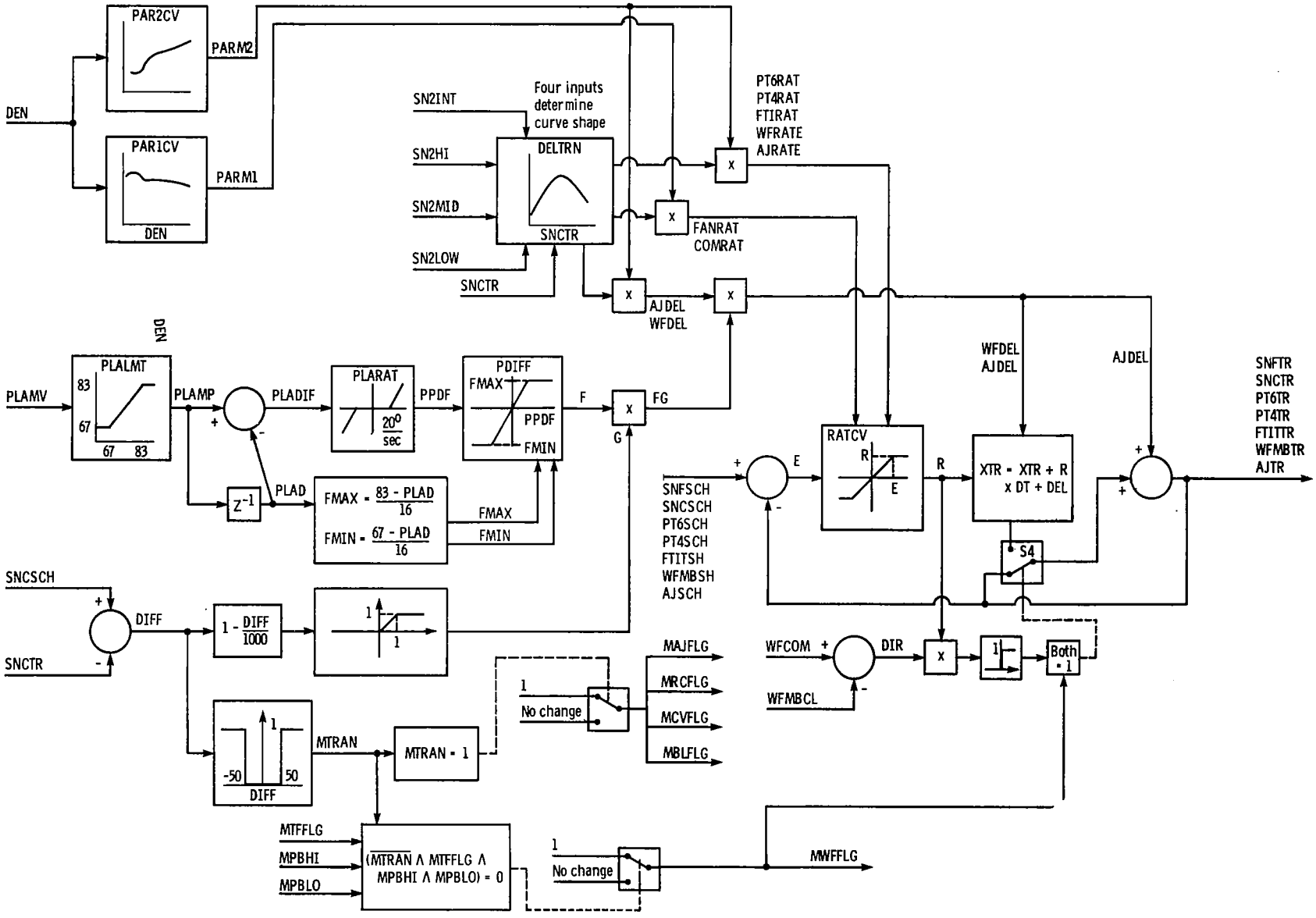


Figure 17. - Multivariable control - transition control.

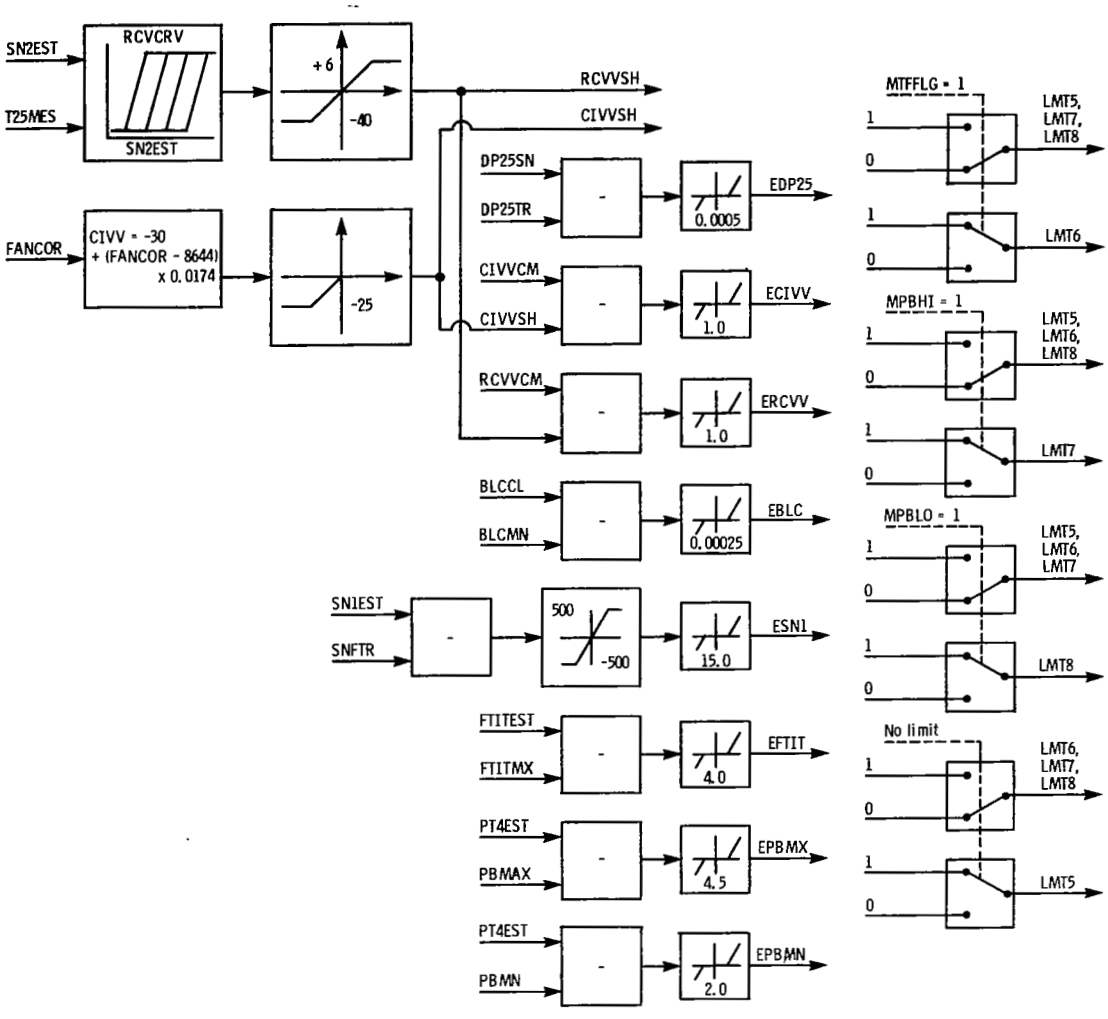


Figure 18. - Multivariable control - integral control.

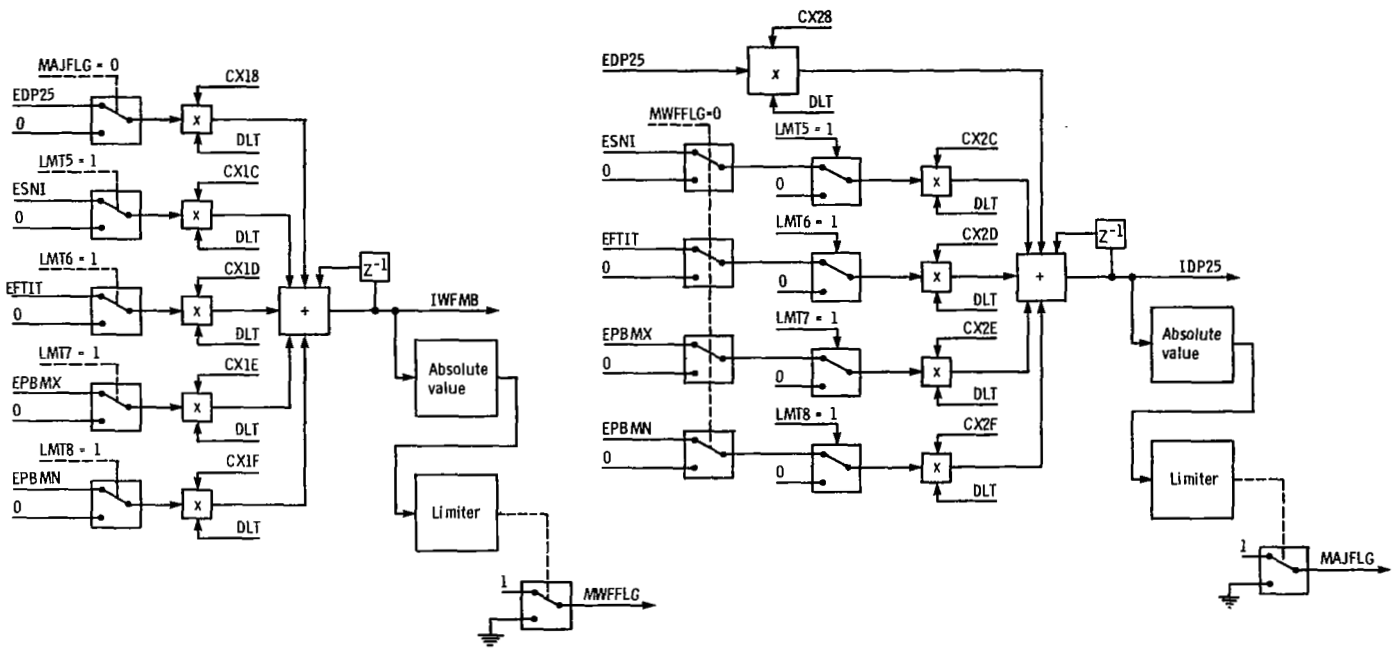


Figure 18. - Continued.

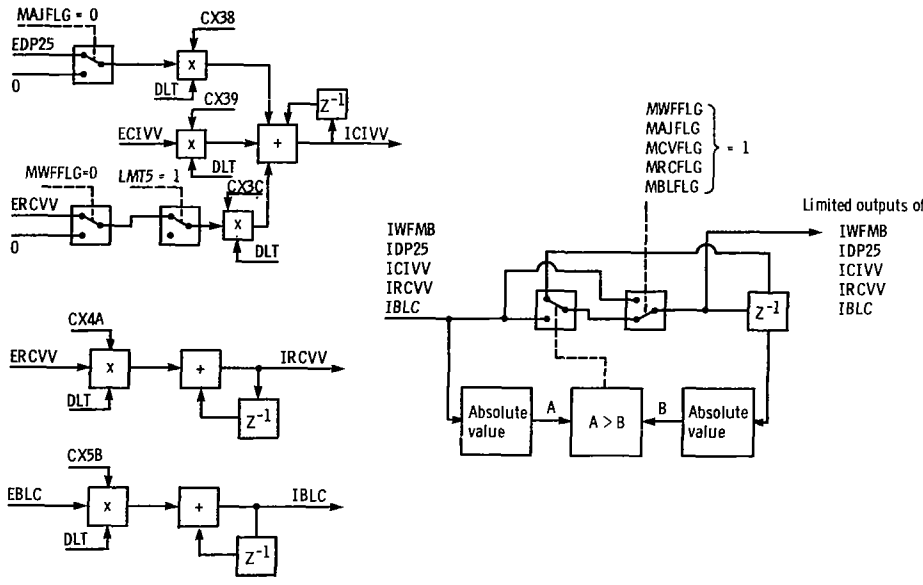


Figure 18. - Concluded.

		Integral output deviations							
		ΔP/P parameter	CIVV	RCVV	Bleed flow	Fan speed	FTIT maximum limit	Combustor pressure minimum limit	Combustor pressure maximum limit
Controls	Fuel flow	X	0	0	0	X	X	X	X
	Nozzle area	X	0	0	0	X	X	X	X
	CIVV	X	X	0	0	X	0	0	0
	RCVV	0	0	X	0	0	0	0	0
	Bleed flow	0	0	0	X	0	0	0	0

Figure 19. - Integral control matrix.

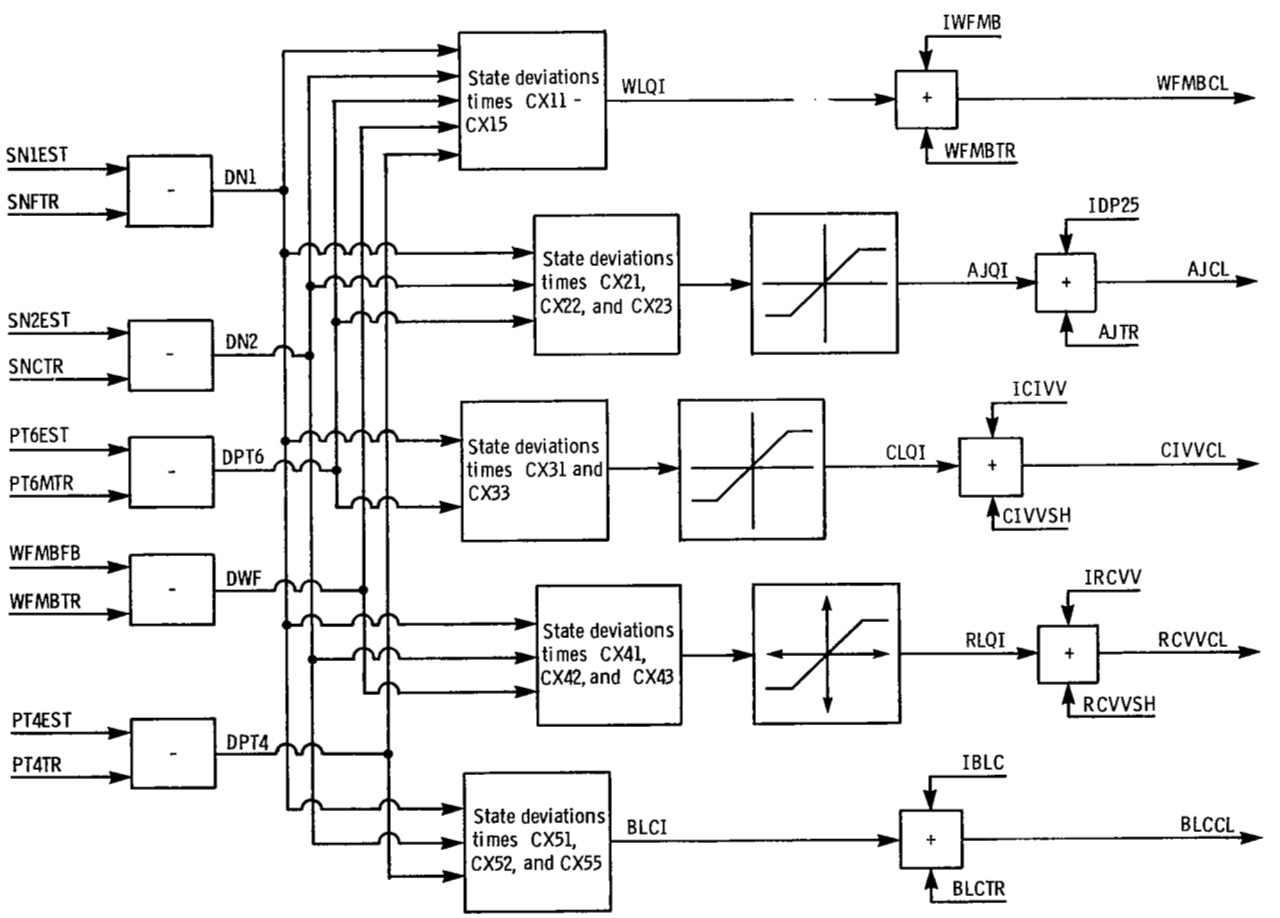


Figure 20. - Multivariable control - LQR control.

		State deviations				
		Fan speed	Compressor speed	Afterburner pressure	Fuel flow	Combustor pressure
Controls	Fuel flow	X	X	X	X	X
	Nozzle area	X	X	X	0	0
	CIVV	X	0	X	0	0
	RCVV	X	X	X	0	0
	Bleed flow	X	X	X	0	X

Figure 21. - LQR gain matrix structure.

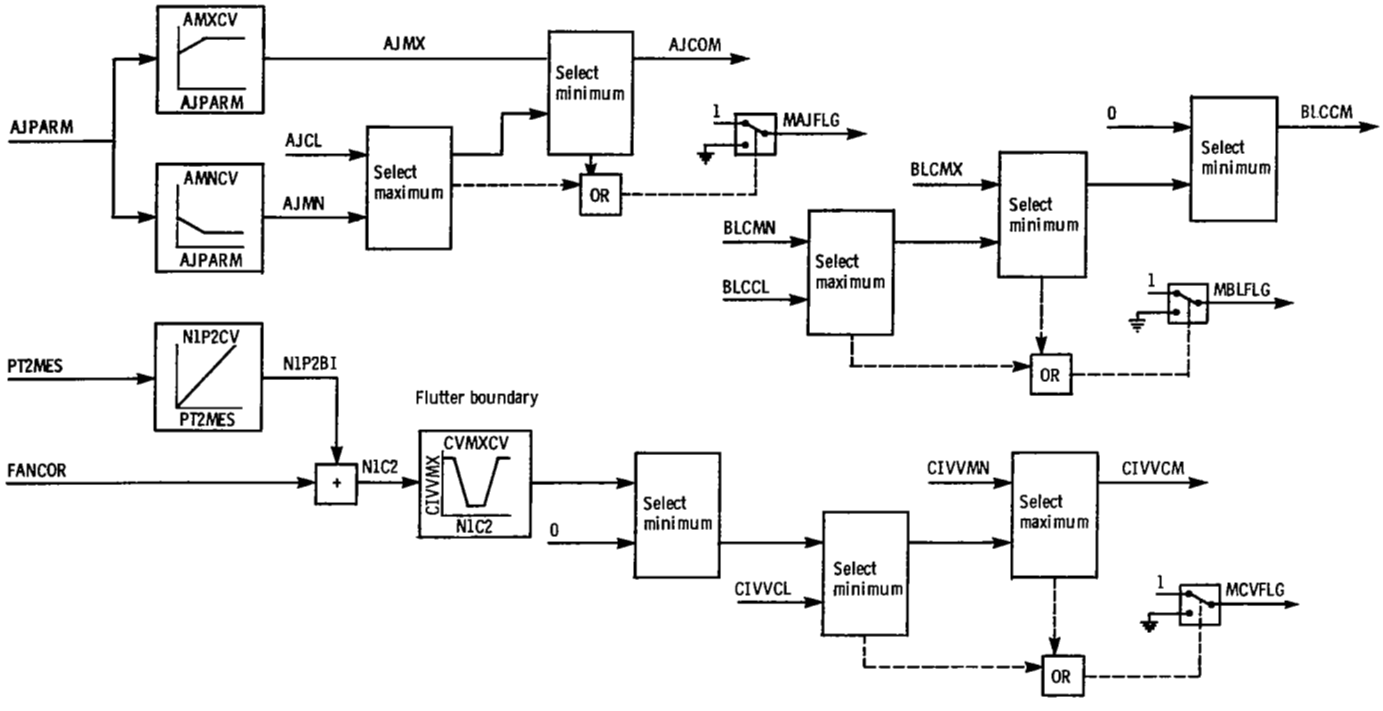


Figure 22. - Multivariable control - engine protection logic.

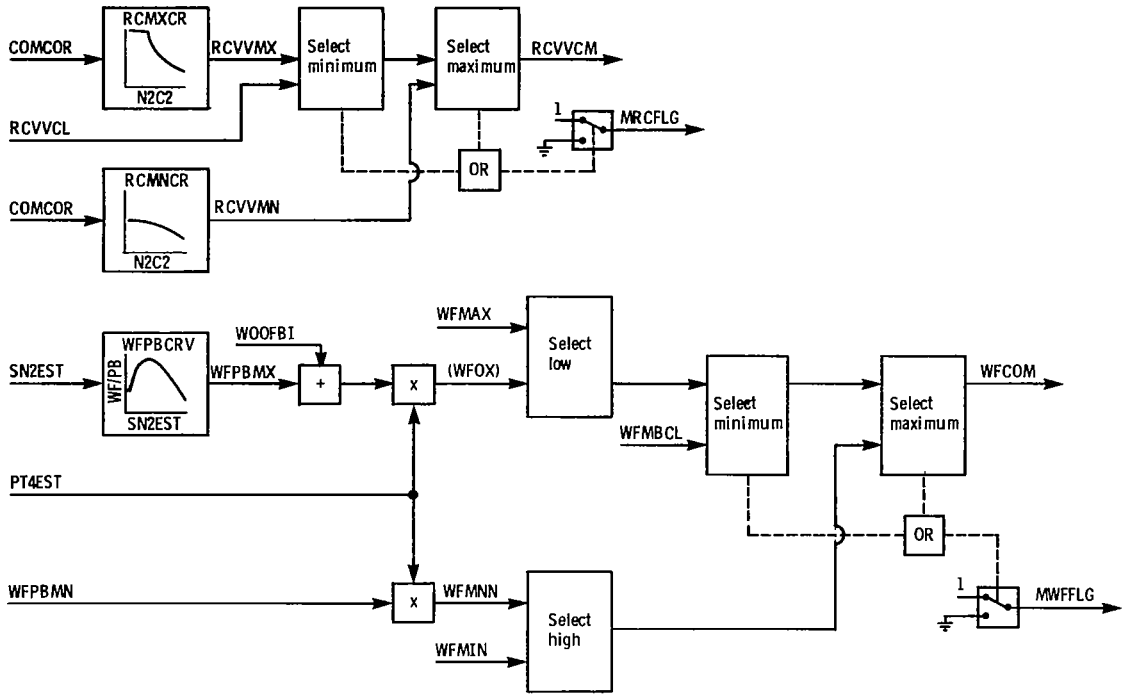


Figure 22. - Concluded.

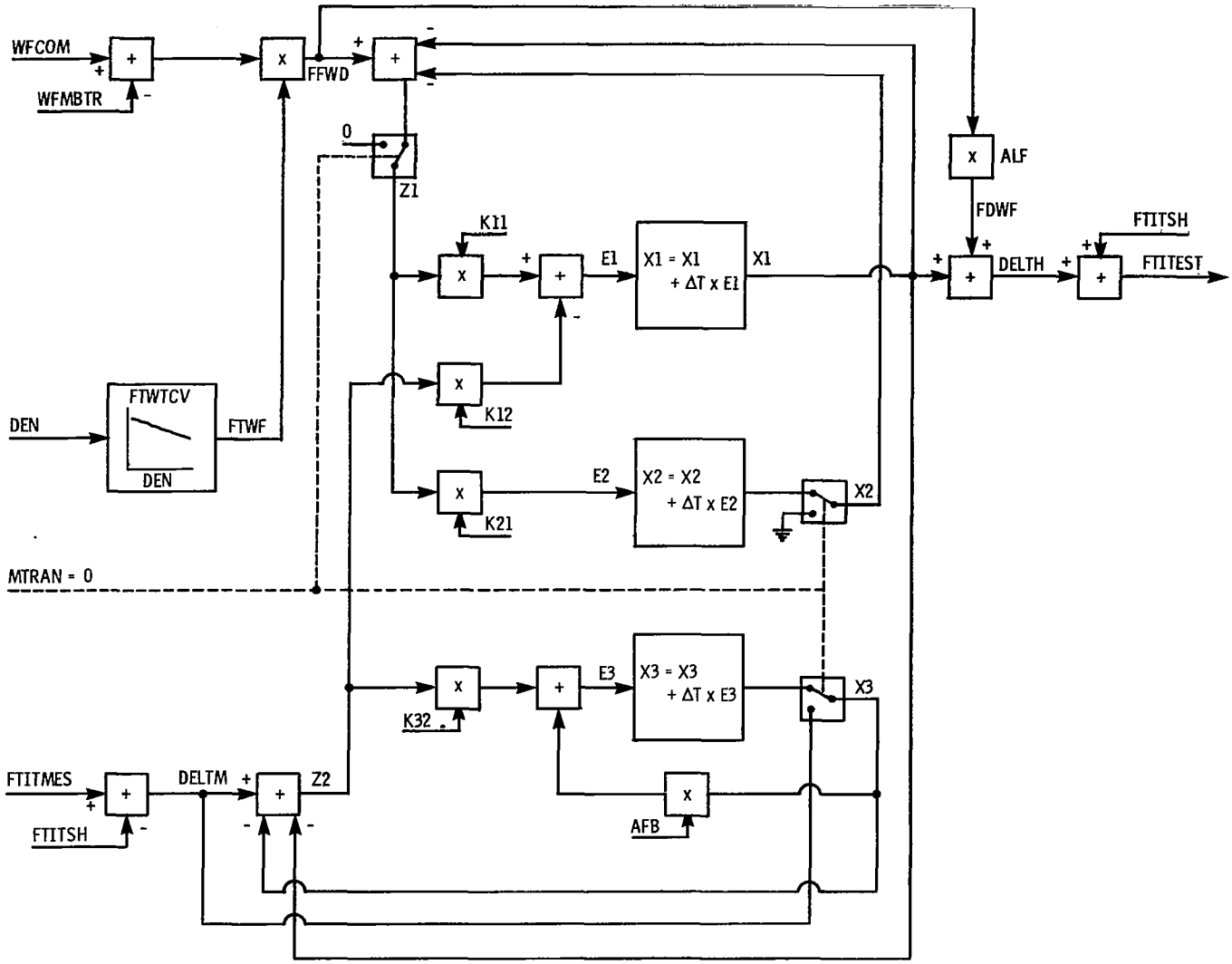


Figure 23. - Multivariable control - FTIT estimator.

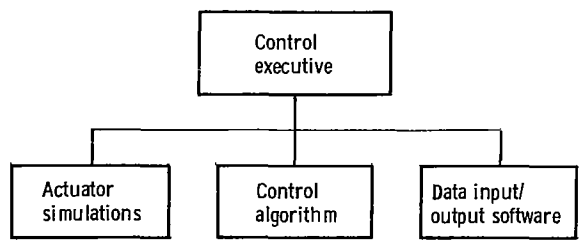


Figure 24. - Multivariable control software configuration (hybrid).

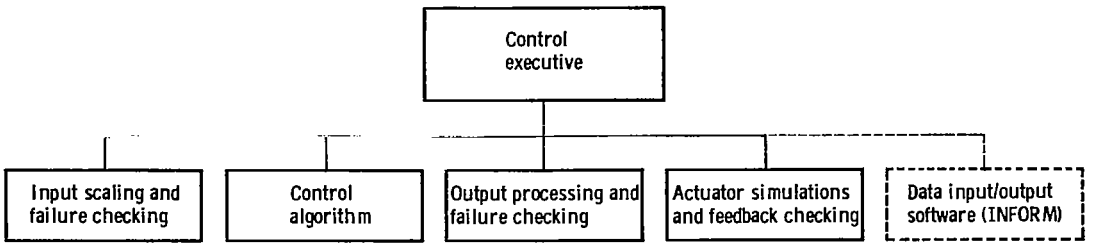


Figure 25. - Multivariable control software configuration (PSL).

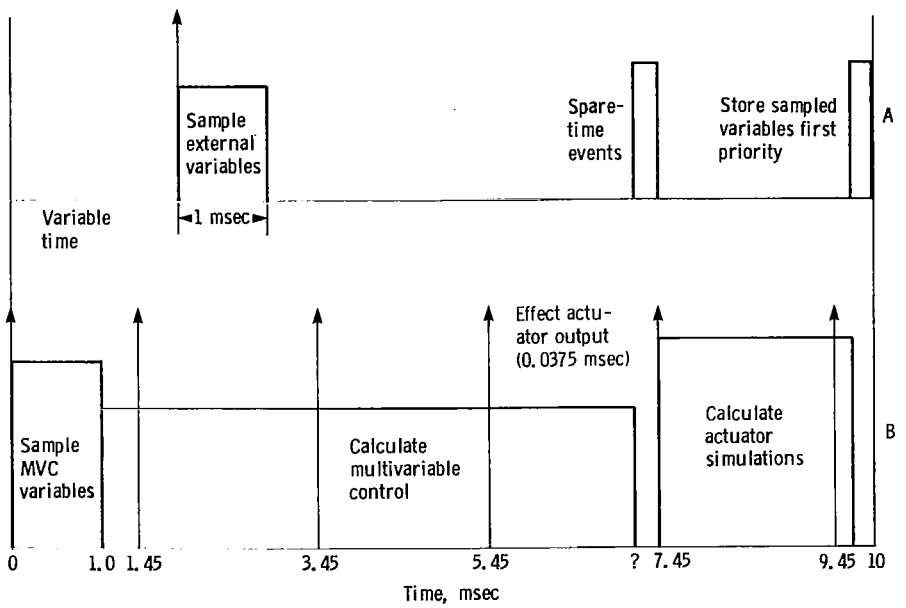


Figure 26. - Multivariable control timing diagram (hybrid).

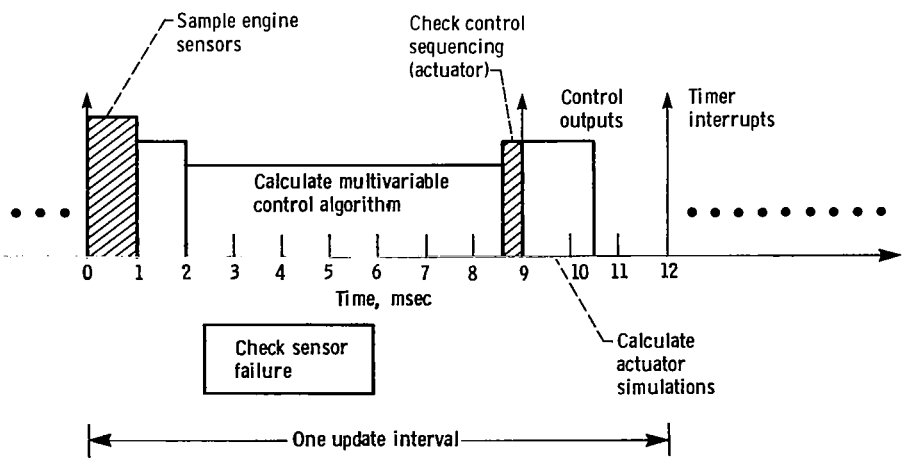


Figure 27. - Multivariable control timing diagram (PSL).

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TP-2231		

4. Title and Subtitle	5. Report Date
F100 MULTIVARIABLE CONTROL SYNTHESIS PROGRAM - COMPUTER IMPLEMENTATION OF THE F100 MULTIVARIABLE CONTROL ALGORITHM	October 1983
	6. Performing Organization Code 505-34-02
7. Author(s) James F. Soeder	8. Performing Organization Report No. E-1496
	10. Work Unit No.
9. Performing Organization Name and Address National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135	11. Contract or Grant No.
	13. Type of Report and Period Covered Technical Paper
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D. C. 20546	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

As turbofan engines become more complex, the development of controls will necessitate the use of multivariable control techniques. This report describes a control developed for the F100-PW-100(3) turbofan engine by using linear quadratic regulator (LQR) theory and other modern multivariable control synthesis techniques. The assembly language implementation of this control on an SEL 810B minicomputer is described. This implementation was then evaluated by using a real-time hybrid simulation of the engine. The control software was modified to run with a real engine in the NASA Lewis Propulsion Systems Laboratory altitude test facility. These modifications, in the form of sensor and actuator failure checks and control executive sequencing, are discussed. Finally recommendations for future control software implementations are presented.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Engine controls Digital controls Control software	Unclassified - unlimited STAR Category 07

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	45	A03

* For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA-Langley, 1983